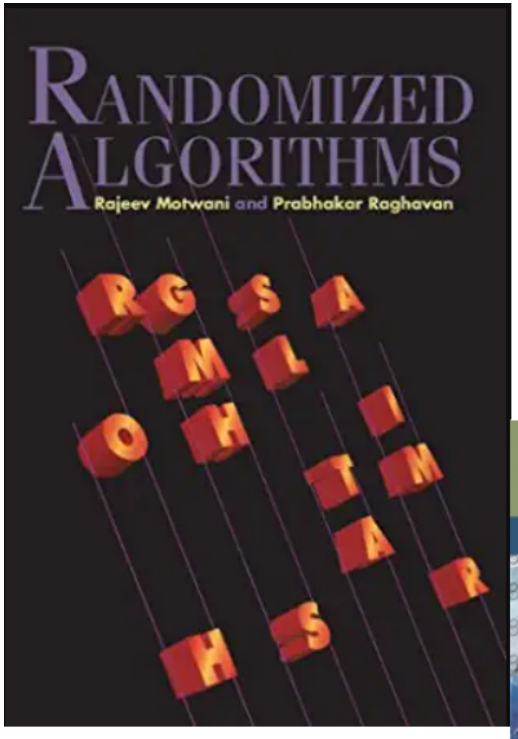
Algoritmos Probabilísticos

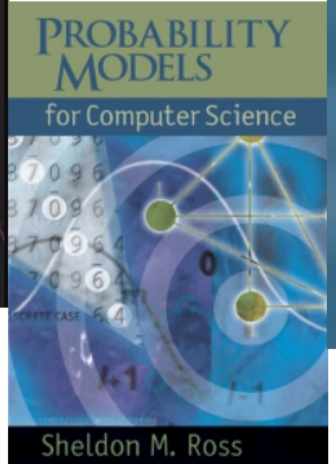
- Algoritmos Probabilísticos, Algoritmos Aleatórios, Algoritmos Randômicos, *Randomized Algorithms*
- São algoritmos que usam probabilidade em sua lógica, ou seja, que fazem escolhas aleatórias.

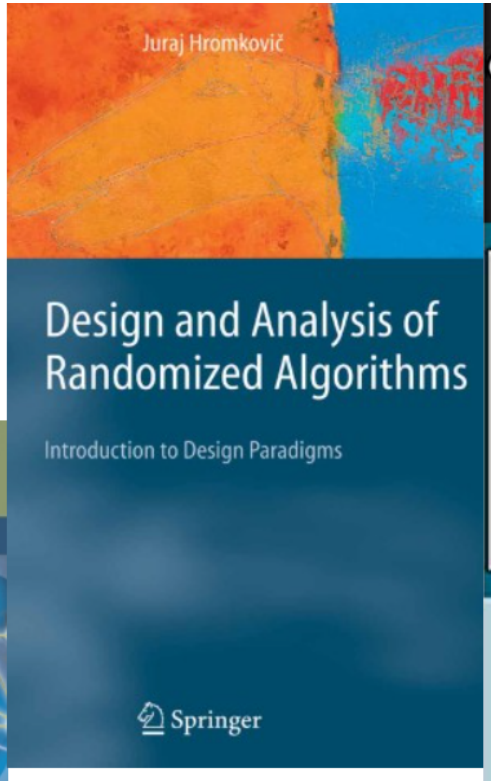
Bibliografia

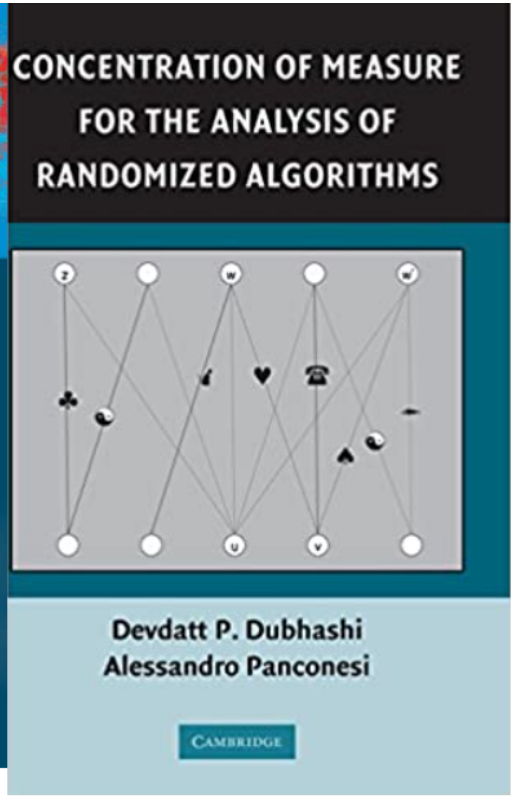
- Motwani, Raghavan, Randomized Algorithms, Cambridge, 1995
- Mitzenmacher, Upfal, Probability and Computing, Cambridge, 2006
- Hromkovic, Design and Analysis of Randomized Algorithms, Springer, 2005
- Sheldon Ross, Probability Models for Computer Science, Academic Press, 2002
- Dubhashi, Panconesi, Concentration of Measure for the Analysis of Randomized Algorithms, Cambridge, 2009

Algoritmos Probabilísticos - Bibliografia









Algoritmos Probabilísticos - Tipos

- **Algoritmo Las Vegas:** Sempre faz o esperado. O tempo do algoritmo depende das escolhas aleatórias (Quick Sort, Seleção).
- **Algoritmo Monte Carlo:** Pode não fazer o esperado, com baixa probabilidade. O tempo do algoritmo pode ou não depender das escolhas aleatórias.

Exemplo: Problema do Popular-busca

- Elem. popular de vetor de tam. n: aparece mais de $n/5$ vezes
- $[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 5, 5, 5]$: O elemento 5 é popular.
- **Problema de Busca:** Dado vetor com popular, encontrar um deles.
- **Algoritmo Las Vegas:** Repita até encontrar: escolhe um elemento aleatório e conta quantas vezes aparece. Se for mais de $n/5$ vezes, retorne esse elemento.
- **Algoritmo Monte Carlo:** Semelhante ao Las Vegas, mas repete no máximo 9 vezes. Retorne um elemento aleatório se não encontrou um popular nas iterações anteriores.

Algoritmos Probabilísticos - Problema Popular-busca

Algoritmo Determinístico p/ Problema Popular-busca

- Elem. popular de vetor de tam. n: aparece mais de $n/5$ vezes
- $[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 5, 5, 5]$: O elemento 5 é popular.
- **Problema de Busca:** Dado vetor com popular, encontrar um deles.
- **Algoritmo Determinístico:** Usa o algoritmo SELEÇÃO do k-ésimo mínimo para $k = n/5$, $2n/5$, $3n/5$ e $4n/5$. Pelo menos um desses 4 elementos será popular, basta contar para descobrir qual.
- $[0, 1, 1, 2, 2, 3, 4, 5, 5, 5, 5, 6, 7, 8, 9]$: O elemento 5 é popular.
- **Tempo de pior caso:** $4 \cdot (n + \text{TempoSeleção}(n)) = O(n)$
- constantes não muito pequenas escondidas na notação: $O(44 \cdot n)$
- Observação: SELEÇÃO tem algoritmo Las-Vegas bem mais rápido.

Algoritmos Probabilísticos - Las Vegas - Popular-busca

Algoritmo Las-Vegas p/ Problema Popular-busca

- Elem. popular de vetor de tam. n: aparece mais de $n/5$ vezes
- $[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 5, 5, 5]$: O elemento 5 é popular.
- **Problema de Busca:** Dado vetor com popular, encontrar um deles.
- **Algoritmo Las Vegas:** Repita até encontrar: escolhe um elemento aleatório e conta quantas vezes aparece. Se for mais de $n/5$ vezes, retorne esse elemento.
- Las Vegas sempre retorna um popular, mas pode demorar.
- Distribuição Geométrica: 5 buscas são suficientes em média
- Tempo Esperado: $O(5n)$
- Mas pode dar tempo bem mais alto: Variância grande

Algoritmos Probabilísticos - Monte Carlo - Popular-busca

Algoritmo Monte-Carlo p/ Problema Popular-busca

- Elem. popular de vetor de tam. n: aparece mais de $n/5$ vezes
- $[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 5, 5, 5]$: O elemento 5 é popular.
- **Problema de Busca:** Dado vetor com popular, encontrar um deles.
- **Algoritmo Monte Carlo:** Semelhante ao Las Vegas, mas repete no máximo 9 vezes. Retorne um elemento aleatório se não encontrou um popular nas iterações anteriores.
- Monte Carlo pode errar e o tempo pode variar: $O(9n)$ pior caso
- Distribuição Geométrica: Probabilidade de errar em 10 tentativas: $(4/5)^{10} < 11\%$
- **Conclusão:** Tempo esperado: $O(3.2 \cdot n)$ com probab acerto 89%
- Tempo não passa de $O(9 \cdot n)$

Algoritmos Probabilísticos - Monte Carlo - Popular-busca

Algoritmo Monte-Carlo-2 p/ Problema Popular-busca

- Elem. popular de vetor de tam. n: aparece mais de $n/5$ vezes
- $[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 5, 5, 5]$: O elemento 5 é popular.
- **Problema de Busca:** Dado vetor com popular, encontrar um deles.
- **Algoritmo Monte-Carlo-2:** Selecione 1000 elementos aleatoriamente e retorne o que aparece mais vezes.
- Monte Carlo pode errar, mas sempre tem mesmo tempo $O(1000)$ constante no pior caso
- Distribuição ????. Probabilidade de errar pequena ????
- **Conclusão:** Tempo pior caso constante e probab acerto grande ????
- **Exercício:** pesquise quanto seria esta probabilidade

Algoritmos Probabilísticos - Popular-decisão

Exemplo: Problema do Popular-decisão

- **Problema de Decisão:** Dado vetor, decidir se ele tem popular.
- **Algoritmo Las Vegas:** ???? (que use realmente probabilidade)
- **Algoritmo Monte Carlo:** Repita no máximo 9 vezes: escolhe um elemento aleatório e conta quantas vezes aparece. Se for mais de $n/5$ vezes, retorne SIM. Se nenhum encontrado, retorne NÃO.
- Monte Carlo pode errar e o tempo pode variar: $O(3.2 \cdot n)$ esperado
- Se NÃO (tem popular), o algoritmo sempre acerta e retorna NÃO.
- Se SIM (tem popular), o algoritmo pode errar.
- Distribuição Geométrica: Probab errar 9 vezes: $(4/5)^9 < 14\%$
- **Conclusão:** Tempo esperado $O(3.2 \cdot n)$ com probab acerto 86%
- Algoritmo *one-sided error true-biased (OK p/ SIM)*: sempre acerta quando diz SIM. Pode errar quando diz NÃO.
- Popular-dec $\in$ classe RP (Randomized Poly) p/ Problemas Decisão
- Outras classes: ZPP (zero error), RP, co-RP, BPP (two sided error)

Próximas aulas

- Resumo das Noções básicas de probabilidade: Independência, Variável aleatória, Esperança, Variância
- Desigualdades clássicas: Markov e Chebyschev
- Aplicação em Algoritmos Probabilísticos interessantes
- Distribuições Discretas: Binomial, Geométrica, HiperGeométrica, Poisson, Binomial Negativa,...
- Aplicações: Problemas do Colecionador de Figurinhas
- Mais Algoritmos Probabilísticos: Quick Sort Aleatório, Mediana Aleatória, MaxSAT
- Desaleatorização: obter algoritmo determinístico (não probabilístico)
- Classificação p/ Problemas de Decisão: ZPP, RP, co-RP, BPP.
- E mais: Amplificação, Cadeias de Markov, outras Desigualdades, outros Algoritmos Probabilísticos,...

Noções básicas de Probabilidade Discreta

Um **espaço de probabilidade discreto** $(\Omega, \mathbb{P})$ é dado por um conjunto Ω (chamado espaço amostral, finito ou infinito enumerável) e uma função de probabilidade $\mathbb{P} : \Omega \to [0, 1]$ tal que $\sum_{\omega \in \Omega} \mathbb{P}(\omega) = 1$.
Subconjuntos do espaço amostral são chamados de **eventos**.

Exemplo: Lançamento de 1 moeda. Espaço amostral {C,K} com $\mathbb{P}(C) = 1/2$ e $\mathbb{P}(K) = 1/2$.

Exemplo: 2 moedas. Espaço amostral {CC,CK, KC, KK} com probabilidade 1/4 para cada.

Exemplo: Lançamento de 1 dado. Espaço amostral {1,2,3,4,5,6} com probabilidade 1/6 para cada.

Exemplo: Duplas em sala com 4 alunos A,B,C,D. Espaço amostral {AB,AC,AD,BC,BD,CD} com probabilidade 1/6 para cada.

Noções básicas de Probabilidade Discreta

Um **espaço de probabilidade discreto** $(\Omega, \mathbb{P})$ é dado por um conjunto Ω (chamado espaço amostral, finito ou infinito enumerável) e uma função de probabilidade $\mathbb{P} : \Omega \to [0, 1]$ tal que $\sum_{\omega \in \Omega} \mathbb{P}(\omega) = 1$.
Subconjuntos do espaço amostral são chamados de **eventos**.

Exemplo: Moeda não-viciada até obter uma coroa. Espaço amostral $\{1, 2, \ldots\}$ do número de lançamentos. Determine função de probab.

Solução: $\mathbb{P}(n) = (\frac{1}{2})^{n-1} \cdot (\frac{1}{2}) = (\frac{1}{2})^n$ (Distribuição Geométrica)
Note que $\sum_{n=1}^{\infty} \mathbb{P}(n) = \sum_{n=1}^{\infty} (\frac{1}{2})^n = \frac{1/2}{1-1/2} = 1$ (soma de PG).

Exemplo: Moeda viciada é lançada até se obter uma coroa, que tem probab. $1/9$. Espaço amostral $\{1, 2, \ldots\}$ do número de lançamentos. Determine a função de probabilidade.

Solução: $\mathbb{P}(n) = (\frac{8}{9})^{n-1} \cdot (\frac{1}{9}) = \frac{1}{8}(\frac{8}{9})^n$ (Distribuição Geométrica)
Note que $\sum_{n=1}^{\infty} \mathbb{P}(n) = \sum_{n=1}^{\infty} \frac{1}{8}(\frac{8}{9})^n = \frac{1/9}{1-8/9} = 1$ (soma de PG).

Noções básicas de Probabilidade Discreta

Exemplo: Moeda viciada até obter 4 coroas. Espaço amostral $\{4, 5, \ldots\}$ do número de lançamentos. Determine função de probab.

Solução: $\mathbb{P}(n) = \binom{n-1}{3}(\frac{1}{9})^4 \cdot (\frac{8}{9})^{n-4}$ (Distr Binomial Negativa)

Exemplo: Uma moeda não-viciada é lançada 5 vezes. Qual é a probabilidade de aparecer apenas uma cara? E 2, 3, 4 ou 5 caras?

Solução: Seja C (cara) e K (coroa).
As possibilidades de lançamentos com 1 cara são:
CKKKK, KCKKK, KKCKK, KKKCK, KKKKC:

$$\begin{aligned}
\mathbb{P}(\text{número de caras} = 0) &= \binom{5}{0}/2^5 = 1/32. \\
\mathbb{P}(\text{número de caras} = 1) &= \binom{5}{1}/2^5 = 5/32. \\
\mathbb{P}(\text{número de caras} = 2) &= \binom{5}{2}/2^5 = 10/32. \quad \text{(Distribuição Binomial)} \\
\mathbb{P}(\text{número de caras} = 3) &= \binom{5}{3}/2^5 = 10/32 \\
\mathbb{P}(\text{número de caras} = 4) &= \binom{5}{4}/2^5 = 5/32. \\
\mathbb{P}(\text{número de caras} = 5) &= \binom{5}{5}/2^5 = 1/32.
\end{aligned}$$

Probabilidade Condicional

Probabilidade de um evento A dado que ocorreu um evento B:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}, \quad \text{se} \quad \mathbb{P}(B) > 0$$

Exemplo: Qual a probabilidade de se obter 3 caras em 5 lançamentos, considerando que nos dois primeiros só teve 1 cara?
As possibilidades são:
CKCCK, CKCKC, CKKCC, KCCCK, KCCKC, KCKCC:

$$\frac{6}{2 \cdot 8} = \frac{3}{8} \quad = \quad \frac{2/4 \cdot 3/8}{16/32}$$

Eventos A e B são **independentes** se $\mathbb{P}(A|B) = \mathbb{P}(A)$.
Conclusão: $\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B)$ e $\mathbb{P}(B|A) = \mathbb{P}(B)$.

Probabilidade Condicional

Probabilidade de um evento A dado que ocorreu um evento B:

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}, \quad \text{se} \quad \mathbb{P}(B) > 0$$

Caso geral: $\mathbb{P}(A \cap B) = \mathbb{P}(A) \cdot \mathbb{P}(B|A)$.
Caso geral: $\mathbb{P}(A \cap B \cap C) = \mathbb{P}(A) \cdot \mathbb{P}(B|A) \cdot \mathbb{P}(C|A, B)$.
Caso mais geral:

$$\mathbb{P}(A_1 \cap A_2 \cap \ldots \cap A_k) = \mathbb{P}(A_1) \cdot \mathbb{P}(A_2|A_1) \cdot \mathbb{P}(A_3|A_1, A_2) \cdot \ldots \cdot \mathbb{P}(A_k|A_1, \ldots, A_{k-1})$$

Amplificação Probabilística da probabilidade de acerto

- Seja $\mathcal{A}$ um algoritmo probabilístico SIM/NÃO *one-sided-error* de tempo polinomial que sempre acerta quando retorna SIM, mas acerta quando retorna NÃO com probabilidade baixa de 1%.
- Repetindo k vezes o algoritmo $\mathcal{A}$ e retornando NÃO só se todas as execuções retornaram NÃO, quanto será a probab de erro? 0.99^k
- Tomando $k = 500$, temos probab de erro $\leq 1\%$.
- O algoritmo probabilístico SIM/NÃO *one-sided-error* obtido desse procedimento é polinomial, sempre acerta quando retorna SIM e acerta quando retorna NÃO com probabilidade alta de 99%.

- Analogamente também vale se $\mathcal{A}$ sempre acerta ao retornar NÃO, mas acerta quando retorna SIM com probabilidade baixa de 1%.
- É mais complicado se o algoritmo probabilístico $\mathcal{A}$ é *two-sided-error*: pode errar o SIM e o NÃO. **Aulas futuras.**

Amplificação Probabilística da probabilidade de acerto

- Seja $\mathcal{A}$ um algoritmo probabilístico SIM/NÃO de tempo polinomial que sempre acerta quando retorna SIM, mas acerta quando retorna NÃO com probabilidade baixa de $\varepsilon > 0$.
- Repetindo k vezes o algoritmo $\mathcal{A}$ e retornando NÃO só se todas as execuções retornaram NÃO, quanto será a probab de erro? $(1-\varepsilon)^k$
- Para $k = \log_{1-\varepsilon} 0.01 = \log_{\frac{1}{1-\varepsilon}} 100$, temos probab erro $\leq 1\%$.
- Exemplo: Para $\varepsilon = 0.01\%$, $k = 50000$ repetições são suficientes.
- O algoritmo probabilístico SIM/NÃO obtido desse procedimento é **polinomial**, sempre acerta quando retorna SIM e acerta quando retorna NÃO com probabilidade alta de 99%.
- Isso porque, quando ε é constante (não depende do tam da instância de $\mathcal{A}$), mesmo que seja muito pequeno, k também é constante.

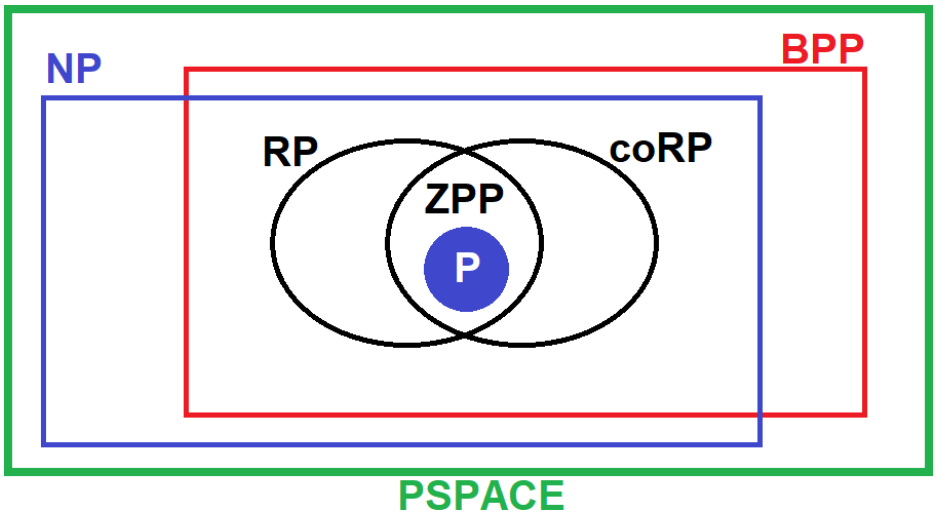
Classes probabilísticas de Problemas de Decisão

Dados $0 \leq s < c \leq 1$, seja $PCP_{c,s}$ o conjunto dos problemas de decisão que possuem algoritmos probabilísticos de tempo polinomial tais que:

- Se instância SIM, $\mathbb{P}$(algoritmo retorna SIM) $\geq c$ (*completeness*)
- Se instância NÃO, $\mathbb{P}$(algoritmo retorna SIM) $\leq s$ (*soundness*)

Classes conhecidas

- ZPP $= PCP_{1,0}$ (Zero erro: possui algoritmo Las Vegas)
- RP $= PCP_{1/2,0}$ (Pode errar dizendo NÃO p/ instância SIM)
- coRP $= PCP_{1,1/2}$ (Pode errar dizendo SIM p/ instância NÃO)
- BPP $= PCP_{2/3,1/3}$ (Pode errar quando retorna SIM ou NÃO)



- P $\subseteq$ ZPP
- ZPP $=$ RP $\cap$ coRP
- BPP $\supseteq$ RP $\cup$ coRP
- P $=$ ZPP ?
- P $=$ BPP ?
- BPP $=$ RP $\cup$ coRP ?

Aplicação: Verificar se tem Popular

- Elem. popular de vetor de tam. n: aparece mais de $n/5$ vezes
- $[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 5, 5, 5]$: O elemento 5 é popular.
- **Problema de Decisão:** Dado vetor, decidir se ele tem popular.
- **Algoritmo Monte Carlo:** Escolhe um elemento aleatório e conta quantas vezes aparece. Se for mais de $n/5$ vezes, retorne SIM. Se nenhum encontrado, retorne NÃO.
- Se NÃO (tem popular), o algoritmo sempre acerta e retorna NÃO.
- Se SIM (tem popular), o algoritmo pode errar com probab $\leq 4/5$.
- **Amplificação:** Repete 21 vezes: erro $(4/5)^{21} < 1\%$
- *One-sided error true biased* (acerta ao dizer SIM): Classe RP.

Aplicação: Verificar Igualdade de Polinômios

Problema: Dados dois polinômios $F(x)$ e $G(x)$ de grau n, decidir se são iguais ou não, considerando $F(x)$ como produto de monômios.
Exemplo: $(x+1)(x-2)(x+3)(x-4) = x^4+7x^2-24$? $(n=4)$

Algoritmo-1: Transformar $F(x)$ p/ forma canônica e comparar com coeficientes de $G(x)$. **Tempo** $\Theta(4+6+8+10+\ldots+2(n-1)) = \Theta(n^2)$

Algoritmo-2: Escolher inteiro r em $\{1,\ldots,100n\}$ aleatório uniforme. Calcular $F(r)$ e $G(r)$ e comparar. Se igual, retornar SIM. Caso contrário, retornar NÃO. **Tempo** $\Theta(n)$. Monte Carlo

Análise Algoritmo-2: Se SIM $(F(x) = G(x))$, o algoritmo acerta. Se NÃO $(F(x) \neq G(x))$ e $F(r) \neq G(r)$, o algoritmo acerta. O algoritmo só erra se for NÃO $(F(x) \neq G(x))$, mas $F(r) = G(r)$. *One-sided error false biased* (acerta ao dizer NÃO): Classe co-RP.

Probabilidade de errar: r deve ser raiz de $F(x) - G(x)$, que tem grau $\leq n$ e no máximo n raizes. Probabilidade de erro $\leq 1/100$

Aplicação: Verificar Igualdade de Polinômios

Problema: Dados dois polinômios $F(x)$ e $G(x)$ de grau n, decidir se são iguais ou não, considerando $F(x)$ como produto de monômios.
Exemplo: $(x+1)(x-2)(x+3)(x-4) \;=\; x^4+7x^2-24$? $\quad (n=4)$

Algoritmo-2: Escolher inteiro r em $\{1,\ldots,100n\}$ aleatório uniforme. Calcular $F(r)$ e $G(r)$ e comparar. Se igual, retornar SIM. Caso contrário, retornar NÃO. **Tempo** $\Theta(n)$. Monte Carlo

Análise Algoritmo-2: Se SIM ($F(x)=G(x)$), o algoritmo acerta. Se NÃO ($F(x)\neq G(x)$) e $F(r)\neq G(r)$, o algoritmo acerta. O algoritmo só erra se for NÃO ($F(x)\neq G(x)$), mas $F(r)=G(r)$. *One-sided error false biased* (acerta ao dizer NÃO): Classe co-RP.

Probabilidade de errar: r deve ser raiz de $F(x)-G(x)$, que tem grau $\leq n$ e no máximo n raizes. Probabilidade de erro $\leq 1/100$

Amplificação: Repetindo k vezes com reposição: $\mathbb{P}(\text{erro}) \leq (1/100)^k$

Aplicação: Verificar produto de matrizes

Problema: Dadas três matrizes $n \times n$ A, B e C, decidir se $A \times B = C$.

Algoritmo-1: Multiplicar $A \times B$ e comparar com C.
Tempo $\Theta(n^3)$ (simples) ou $\Theta(n^{2.37})$ (sofisticado)

Algoritmo-2: Escolher vetor $r = (r_1, \ldots, r_n)$ em $\{0,1\}^n$ aleatório uniforme. Calcular $A \cdot (B \cdot r)$ e $C \cdot r$ e comparar. Se igual, retornar SIM. Caso contrário, retornar NÃO. **Tempo** $\Theta(n^2)$. Monte Carlo

Análise Algoritmo-2: Se SIM ($AB = C$), o algoritmo acerta.
Se NÃO ($AB \neq C$) e $A \cdot (B \cdot r) \neq C \cdot r$, o algoritmo acerta.
O algoritmo só erra se for NÃO ($AB \neq C$), mas $A \cdot (B \cdot r) = C \cdot r$.
One-sided error false biased (acerta ao dizer NÃO): Classe co-RP.

Aplicação: Verificar produto de matrizes

Problema: Dadas três matrizes $n \times n$ A, B e C, decidir se $A \times B = C$.

Algoritmo-2: Escolher vetor $r = (r_1, \ldots, r_n)$ em $\{0,1\}^n$ aleatório uniforme. Calcular $A \cdot (B \cdot r)$ e $C \cdot r$ e comparar. Se igual, retornar SIM. Caso contrário, retornar NÃO. **Tempo** $\Theta(n^2)$. Monte Carlo

Análise Algoritmo-2: Se SIM ($AB = C$), o algoritmo acerta.
Se NÃO ($AB \neq C$) e $A \cdot (B \cdot r) \neq C \cdot r$, o algoritmo acerta.
O algoritmo só erra se for NÃO ($AB \neq C$), mas $A \cdot (B \cdot r) = C \cdot r$.
One-sided error false biased (acerta ao dizer NÃO): Classe co-RP.

Probabilidade de errar: Seja $D = AB - C \neq 0$. Logo D tem valor $\neq 0$ (para simplificar, suponha que $d_{1,1} \neq 0$).

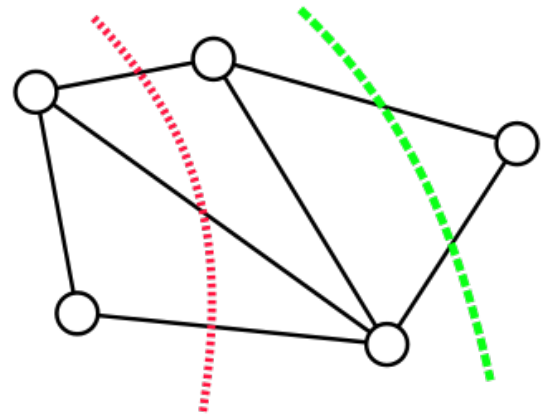
$$A \cdot (B \cdot r) = C \cdot r \Rightarrow D \cdot r = 0 \Rightarrow \sum_{k=1}^{n} d_{1,k} \cdot r_k = 0 \Rightarrow r_1 = -\sum_{k=2}^{n} \frac{d_{1,k} \cdot r_k}{d_{1,1}}$$

$\mathbb{P}(r_1 \text{ ser este valor} \mid r_2, \ldots, r_n) \leq 1/2$. Logo $\mathbb{P}(\text{erro}) \leq 1/2$.

Amplificação: Repetindo k vezes com reposição: $\mathbb{P}(\text{erro}) \leq (1/2)^k$

Algoritmo de Karger'93 para Corte Mínimo

Min-Cut (otimização): Dado um grafo simples G, obter o menor número de arestas cuja remoção desconecta o grafo.



Algoritmo de Fluxo Máximo: Aplicar Ford-Fulkerson (ou outro) p/ cada par (s, t) de vértices, pondo capacidade 1 e direção nos dois sentidos para cada aresta. Ok pelo Teorema max-flow min-cut. Tempo $O(n^5)$.

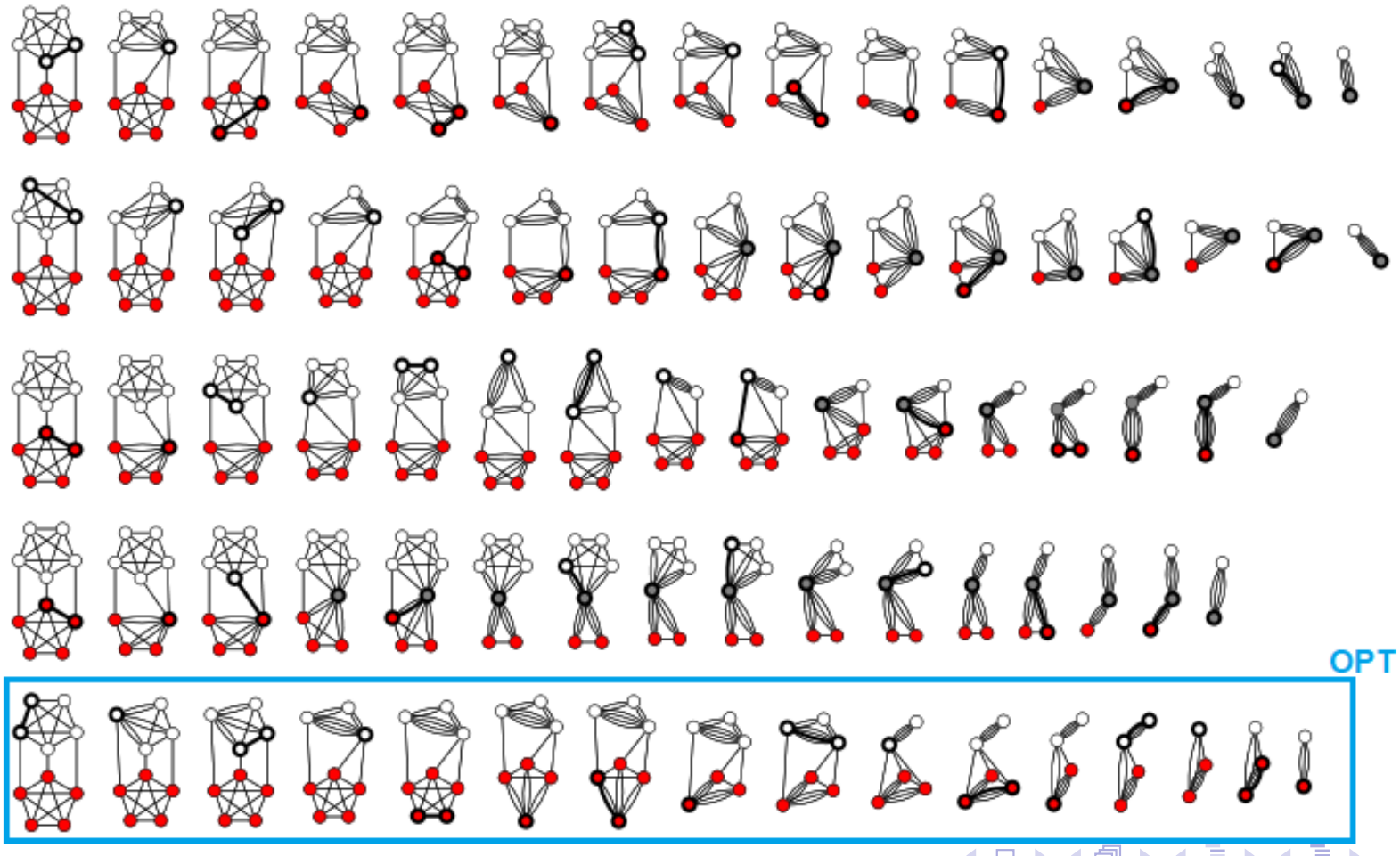
Algoritmo de Gabow'95: matroides, mais complicado. Tempo $O(n^3)$.

Algoritmo de Karger'93: Probabilístico, mais simples e mais rápido. Tempo $O(n^2 \log^3 n)$.

Algoritmo de Karger'93 para Corte Mínimo

Min-Cut (otimização): Dado um grafo simples G, obter o menor número de arestas cuja remoção desconecta o grafo.

Algoritmo de Karger'93: Escolhe uma aresta aleatoriamente de modo uniforme e a contrai. Repete até sobrar apenas 2 vértices.



Algoritmo de Karger'93 para Corte Mínimo

Algoritmo de Karger'93: Escolhe uma aresta aleatoriamente de modo uniforme e a contrai. Repete até sobrar apenas 2 vértices.

Teorema: Algoritmo encontra Corte Mínimo com prob. $\geq \frac{2}{n(n-1)}$.

Prova: Seja C um corte mínimo com k arestas. Vamos calcular a probabilidade de encontrar C. O algoritmo não pode contrair nenhuma aresta de C. Como C é mínimo, todo vértices tem grau $\geq k$. Logo $m \geq nk/2$ (número arestas).

$$\mathbb{P}(1^o \text{ iteração OK}) \geq 1 - \frac{k}{nk/2} = 1 - \frac{2}{n} = \frac{n-2}{n}$$

Vértices restantes mantém grau $\geq k \Rightarrow (n-1)k/2$ arestas.

$$\mathbb{P}(2^o \text{ iteração OK}) \geq 1 - \frac{k}{(n-1)k/2} = 1 - \frac{2}{n-1} = \frac{n-3}{n-1}$$

Vértices restantes mantém grau $\geq k \Rightarrow (n-2)k/2$ arestas.

$$\mathbb{P}(3^o \text{ iteração OK}) \geq 1 - \frac{k}{(n-2)k/2} = 1 - \frac{2}{n-2} = \frac{n-4}{n-2}$$

Algoritmo de Karger'93 para Corte Mínimo

Algoritmo de Karger'93: Escolhe uma aresta aleatoriamente de modo uniforme e a contrai. Repete até sobrar apenas 2 vértices.

Teorema: Algoritmo encontra Corte Mínimo com prob. $\geq \frac{2}{n(n-1)}$.

Prova: Seja C um corte mínimo com k arestas. Vamos calcular a probabilidade de encontrar C. O algoritmo não pode contrair nenhuma aresta de C. Como C é mínimo, todo vértices tem grau $\geq k$. Logo $m \geq nk/2$ (número arestas).

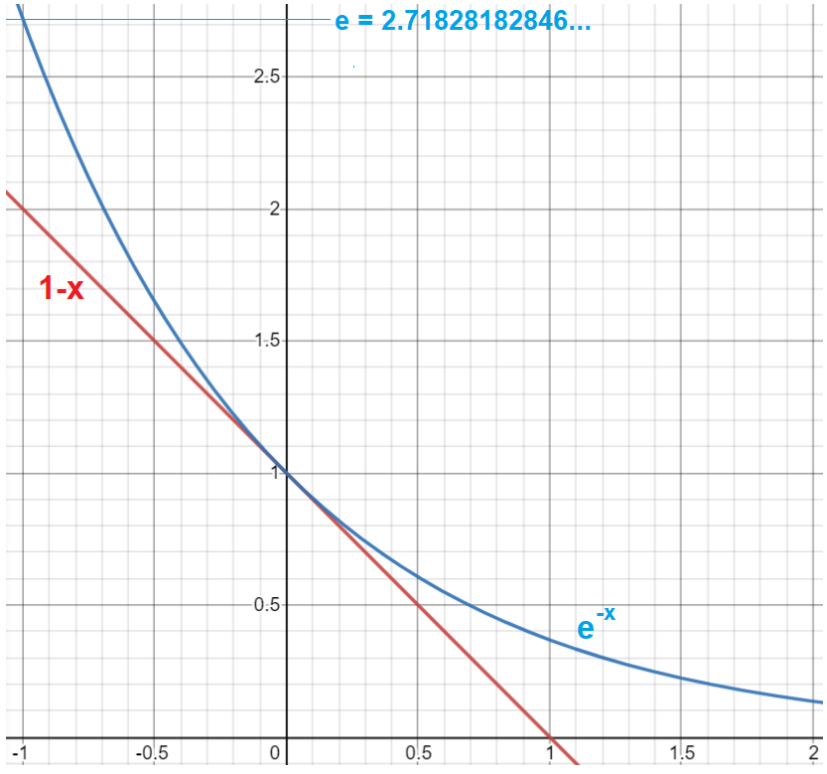
$$\mathbb{P}(\text{tudo OK}) \;\geq\; \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1} \;=\; \frac{2 \cdot 1}{n(n-1)} \qquad \square$$

Repetindo $3n(n-1)$ vezes e retornando o menor corte obtido, qual a probabilidade de um corte mínimo C não ser encontrado?

$$\left(1 - \frac{2}{n(n-1)}\right)^{3n(n-1)} \;\leq\; e^{-6} \;<\; 1\%,$$

pois $1 - x \leq e^{-x}$ para todo x (especialmente x próximo de zero).

Algoritmo de Karger'93 para Corte Mínimo

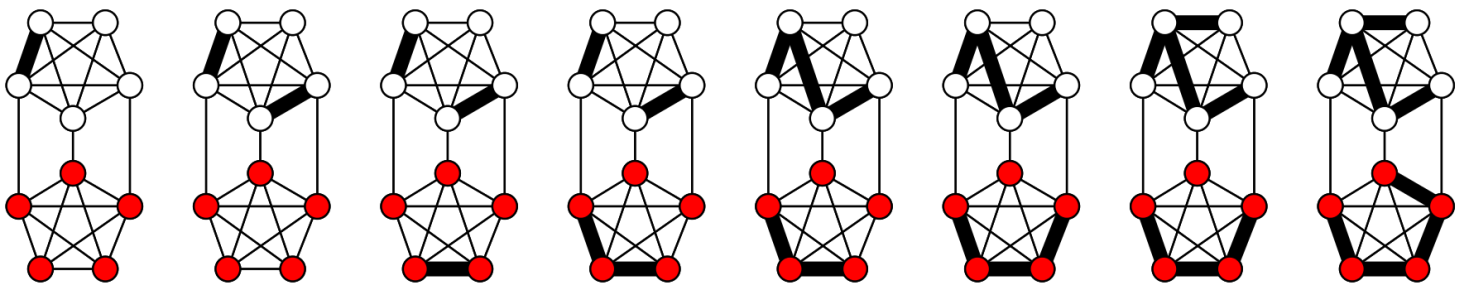


Repetindo $n(n-1)\ln n$ vezes e retornando o menor corte obtido, qual a probabilidade de um corte mínimo C não ser encontrado?

$$\left(1-\frac{2}{n(n-1)}\right)^{n(n-1)\ln n} \leq e^{-2\ln n} = \frac{1}{n^2}.$$

Algoritmo de Karger'93 para Corte Mínimo

Tempo do algoritmo: $O(n^2)$ vezes o tempo do processo de contração, que é similar à execução do Algoritmo de Kruskal (para árvore geradora mínima com pesos aleatórios) até sobrarem 2 componentes conexas.
Tempo total: $O(n^4)$. Pode ser melhorado para tempo $O(n^2 \log^3 n)$.



Observação final: Os algoritmos probabilísticos vistos nesta disciplina obtém tempos melhores que os algoritmos determinísticos tanto para problemas de busca (popular) e de decisão (igualdade de polinômios, produto de matrizes) como otimização (corte mínimo). Porém, não se conhece algoritmo probabilístico polinomial eficiente que resolva um problema NP-Difícil. Isso está relacionado às questões P=ZPP? e P=BPP? já mencionadas. Ou seja, não se espera com o uso de probabilidade obter algoritmos polinomiais para problemas NP-Difíceis, mas sim obter algoritmos mais rápidos do que os existentes com ajuda da processos aleatórios.

Variável Aleatória Discreta

Uma **variável aleatória (v.a.)** de um espaço de probabilidade discreto $(\Omega, \mathbb{P})$ é uma função $X : \Omega \to \mathbb{R}$ que associa um valor real a cada elemento do espaço amostral Ω.

Exemplo: 3 moedas; v.a. X conta o número de caras.
$X(KKC) = 1$, $X(CCK) = 2$.

Probabilidade de uma v.a.: Escrevemos $\mathbb{P}(X \in S)$ para $S \subseteq \mathbb{R}$:

$$\mathbb{P}(X \in S) \;=\; \mathbb{P}(X^{-1}(S)) \;=\; \sum_{\omega \in X^{-1}(S)} \mathbb{P}(\omega).$$

Exemplo: 3 moedas; v.a. X conta o número de caras.
$\mathbb{P}(X \geq 2) = \mathbb{P}(\{CCK, CKC, KCC, CCC\}) = 4 \cdot \frac{1}{8} = \frac{1}{2}$.

Observação: A maioria das variáveis aleatórias usadas aqui serão do tipo $X : \Omega \to \mathbb{N}$. Nesse caso, $\sum_{x=0}^{\infty} \mathbb{P}(X = x) = 1$

Distribuição de Probabilidade Conjunta

Dadas duas v.a. X e Y de um espaço de probabilidade discreto $(\Omega, \mathbb{P})$, escrevemos $\mathbb{P}(X \in A, Y \in B)$ como a probabilidade de $X \in A$ e $Y \in B$.

X e Y são **independentes** se, para quaisquer conjuntos A e B: $\mathbb{P}(X \in A, Y \in B) = \mathbb{P}(X \in A) \cdot \mathbb{P}(Y \in B)$.

Exemplo: 3 moedas. v.a.'s X (num caras), Y (num coroas), Z (num caras nos 2 primeiros lançamentos) e W (num caras no último).

$$\mathbb{P}(X = 2, Y = 0) = 0 \neq \frac{3}{8} \cdot \frac{1}{8}$$ X e Y não são independentes

$$\mathbb{P}(X = 3, Z = 1) = 0 \neq \frac{1}{8} \cdot \frac{2}{4}$$ X e Z não são independentes

$$\mathbb{P}(Z = 0, W = w) = \frac{1}{8} = \frac{1}{4} \cdot \frac{1}{2}$$ Analogamente para $Z = 2$

$$\mathbb{P}(Z = 1, W = w) = \frac{1}{4} = \frac{1}{2} \cdot \frac{1}{2}$$ Logo Z e W são independentes

Valor esperado, Esperança ou Média $\mathbb{E}(X)$

v.a. discreta X: $$\mathbb{E}(X) = \sum_x x \cdot \mathbb{P}(X = x)$$

Média dos valores de X, ponderada pela probabilidade de cada valor.

Ex: valor X de 1 dado: $\mathbb{E}(X) = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + 3 \cdot \frac{1}{6} + \ldots + 6 \cdot \frac{1}{6} = 3.5$

Ex: num caras em 3 moedas: $\mathbb{E}(X) = 0 \cdot \frac{1}{8} + 1 \cdot \frac{3}{8} + 2 \cdot \frac{3}{8} + 3 \cdot \frac{1}{8} = 1.5$

v.a.'s discretas X e Y: $$\mathbb{E}(XY) = \sum_{x,y} x \cdot y \cdot \mathbb{P}(X = x, Y = y)$$

Se X e Y são independentes, então: (a volta nem sempre vale)

$$\mathbb{E}(XY) = \sum_x x \sum_y y \cdot \mathbb{P}(X = x) \cdot \mathbb{P}(Y = y) = \mathbb{E}(X) \cdot \mathbb{E}(Y)$$

Ex: $\mathbb{E}(XY) = 0 \cdot \frac{1}{8} + 2 \cdot \frac{3}{8} + 2 \cdot \frac{3}{8} + 0 \cdot \frac{1}{8} = 1.5 \neq 1.5^2 = \mathbb{E}(X) \cdot \mathbb{E}(Y)$

Linearidade da Esperança $\mathbb{E}(X)$

v.a. discreta X, função $g : \mathbb{R} \to \mathbb{R}$: $\mathbb{E}(g(X)) = \sum_x g(x) \cdot \mathbb{P}(X = x)$

$$\mathbb{E}(aX + b) = \sum_x (ax + b) \cdot \mathbb{P}(X = x) = a \cdot \mathbb{E}(X) + b$$

Linearidade da Esperança: $\mathbb{E}(X + Y) = \mathbb{E}(X) + \mathbb{E}(Y)$

Prova:

$$\mathbb{E}(X + Y) = \sum_x \sum_y (x + y) \cdot \mathbb{P}(X = x, Y = y) =$$

$$= \sum_x x \cdot \sum_y \mathbb{P}(X = x, Y = y) + \sum_y y \cdot \sum_x \mathbb{P}(X = x, Y = y)$$

$$= \sum_x x \cdot \mathbb{P}(X = x) + \sum_y y \cdot \mathbb{P}(Y = y) = \mathbb{E}(X) + \mathbb{E}(Y) \quad \square$$

Observação: Para variáveis contínuas, vale o mesmo (trocando somatórios por integrais).

Método Probabilístico: subgrafo bipartido grande

Método para provar resultados combinatórios teóricos usando probabilidade. Ideias aplicáveis a algoritmos probabilísticos (Paul Erdős).

Teorema: Todo grafo $G = (V, E)$ possui um subgrafo bipartido com pelo menos $m/2$ (metade) das arestas, onde $m = |E|$ e $n = |V|$.
Prova:

- Seja A um subconjunto aleatório de vértices: $\mathbb{P}(v \in A) = p = 1/2$.
- Seja $B = V - A$ e seja X a v.a. do num arestas entre A e B.
- $X = \sum_{uv \in E} X_{uv}$, onde $X_{uv} = 1$, se uv está entre A e B, e 0 c.c.
- $\mathbb{E}(X_{uv}) = 1 \cdot (p \cdot (1-p) + (1-p) \cdot p) = \frac{1}{2}$
- $\mathbb{E}(X) = m/2$. Logo existe subgrafo bipartido com $m/2$ arestas. □

Algoritmo Probabilístico: Seja A inicialmente vazio. Para cada vértice de G, coloque-o em A com probabilidade $1/2$. Seja $B = V - A$. Remova todas as arestas entre vértices de A e remova todas as arestas entre vértices de B. O subgrafo resultante é bipartido com número esperado de arestas igual a $m/2$.

Ainda não fizemos conta da probabilidade desse algoritmo errar, produzindo um subgrafo bipartido com menos da metade do número de arestas. Próximas aulas.

Método Probabilístico: subgrafo bipartido grande c/ pesos

Método para provar resultados combinatórios teóricos usando probabilidade. Ideias aplicáveis a algoritmos probabilísticos (Paul Erdős).

Teorema: Todo grafo $G = (V, E)$ com pesos nas arestas tem subgrafo bipartido com pelo menos metade do peso total $W = \sum_{uv \in E} w_{uv}$.
Prova:

- Seja A um subconjunto aleatório de vértices: $\mathbb{P}(v \in A) = p = 1/2$.
- Seja $B = V - A$ e X a v.a. do peso total das arestas entre A e B.
- $X = \sum_{uv \in E} X_{uv}$, onde $X_{uv} = w_{uv}$, se uv está entre A e B, e 0 c.c.
- $\mathbb{E}(X_{uv}) = w_{uv} \cdot (p \cdot (1-p) + (1-p) \cdot p) = w_{uv}/2$
- $\mathbb{E}(X) = W/2$. Logo há subgrafo bipartido com peso total $W/2$. $\square$

Algoritmo Probabilístico: Seja A inicialmente vazio. P/ cada vértice de G, coloque-o em A com probabilidade $1/2$. Seja $B = V - A$. Remova todas as arestas entre vértices de A e remova todas as arestas entre vértices de B. O subgrafo gerado é bipartido com peso esperado $W/2$.

Método Probabilístico: conjunto independente grande

Método para provar existência de estruturas combinatórias usando probabilidade. Ideias aplicáveis a algoritmos probabilísticos (Paul Erdős).

Teorema: Todo grafo $G = (V, E)$ com n vértices e $m \leq nd/2$ arestas (para algum $d \geq 1$, por exemplo grau máximo) possui $\alpha(G) \geq n/2d$, onde $\alpha(G)$ é o tamanho do conjunto independente máximo de G (subconjunto de vértices sem arestas entre si).
Prova:

- Seja A um subconjunto aleatório de vértices: $\mathbb{P}(v \in A) = p$ a ser definido.
- Seja v.a. $X = |A|$ ($\mathbb{E}(X) = np$) e Y o num arestas em $G[A]$.
- Logo $Y = \sum_{uv \in E} Y_{uv}$, onde $Y_{uv} = 1$, se $u, v \in A$, e 0 c.c.
- $\mathbb{E}(Y) = \sum_{uv \in E} \mathbb{E}(Y_{uv}) = m \cdot p^2 \leq (nd/2)p^2$.
- $\mathbb{E}(X - Y) \geq np - (nd/2)p^2$ (removendo um vértice para cada aresta).
- Tomando $p = 1/d$ para maximização: $\mathbb{E}(X - Y) \geq n/2d$ □

Algoritmo Probabilístico: Seja A inicialmente vazio. Para cada vértice de G, coloque-o em A com probabilidade $1/d$. Remova todos os vértices fora de A. Iterativamente, remova o vértice com maior grau até o subgrafo resultante não ter arestas. Tamanho esperado $n/2d$.

Variância $\mathbb{V}ar(X)$, Desvio padrão $\sigma(X)$ e Covariância

Medidas para avaliar o quanto uma v.a. X desvia de sua média.

$$Var(X) = \mathbb{E}\Big((X - \mathbb{E}(X))^2\Big) = \mathbb{E}(X^2) - \mathbb{E}^2(X). \qquad \sigma(X) = \sqrt{\mathbb{V}ar(X)}$$

$$\mathbb{V}ar(a \cdot X + b) \;=\; a^2 \cdot \mathbb{V}ar(X) \quad \text{e} \quad \sigma(a \cdot X + b) = a \cdot \sigma(X)$$

Ex: valor X de 1 dado:
$\mathbb{E}(X^2) = 1^2 \cdot \frac{1}{6} + 2^2 \cdot \frac{1}{6} + 3^2 \cdot \frac{1}{6} + \ldots + 6^2 \cdot \frac{1}{6} = \frac{91}{6} = 15.2.$
$\mathbb{V}ar(X) = 15.2 - 3.5^2 = 2.92. \qquad \sigma(X) = \sqrt{\mathbb{V}ar(X)} = 1.71.$

Covariância de v.a. X e Y: $\mathbb{C}ov(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X) \cdot \mathbb{E}(Y)$.
Logo, se X e Y são independentes, então $\mathbb{C}ov(X, Y) = 0$. (a volta não vale)

Propriedades da Covariância: $\mathbb{C}ov(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X) \cdot \mathbb{E}(Y)$

- $\mathbb{C}ov(X, X) = Var(X)$
- $\mathbb{C}ov(X, Y) = Cov(Y, X)$
- $\mathbb{C}ov(a \cdot X, Y) = a \cdot \mathbb{C}ov(X, Y)$
- $\mathbb{C}ov(X, Y + Z) = Cov(X, Y) + Cov(X, Z)$

Variância $\mathbb{V}ar(X)$ e Covariância $\mathbb{C}ov(X, Y)$

Propriedades da Covariância: $\mathbb{C}ov(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X) \cdot \mathbb{E}(Y)$

- $\mathbb{C}ov(X, X) = Var(X)$
- $\mathbb{C}ov(X, Y) = Cov(Y, X)$
- $\mathbb{C}ov(a \cdot X, Y) = a \cdot \mathbb{C}ov(X, Y)$
- $\mathbb{C}ov(X, Y + Z) = Cov(X, Y) + Cov(X, Z)$

$$\mathbb{C}ov\left(\sum_{i=1}^{n} X_i \ , \ \sum_{j=1}^{m} Y_j\right) \ = \ \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{C}ov(X_i, Y_j)$$

$$\mathbb{V}ar\left(\sum_{i=1}^{n} X_i\right) \ = \ \mathbb{C}ov\left(\sum_{i=1}^{n} X_i \ , \ \sum_{j=1}^{n} X_j\right) \ = \ \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{C}ov(X_i, X_j) \ =$$

$$= \sum_{i=1}^{n} \mathbb{C}ov(X_i, X_i) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathbb{C}ov(X_i, X_j) \ = \ \sum_{i=1}^{n} \mathbb{V}ar(X_i) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathbb{C}ov(X_i, X_j)$$

Se $X_1, \ldots, X_n$ são independentes, então: $\mathbb{V}ar(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathbb{V}ar(X_i)$.

Variância, Desvio Padrão e Covariância

Propriedades da Covariância: $\mathbb{C}ov(X, Y) = \mathbb{E}(XY) - \mathbb{E}(X) \cdot \mathbb{E}(Y)$

- $\mathbb{C}ov(X, X) = Var(X)$
- $\mathbb{C}ov(X, Y) = Cov(Y, X)$
- $\mathbb{C}ov(a \cdot X, Y) = a \cdot \mathbb{C}ov(X, Y)$
- $\mathbb{C}ov(X, Y + Z) = Cov(X, Y) + Cov(X, Z)$

$$\mathbb{C}ov\left(\sum_{i=1}^{n} X_i \ , \ \sum_{j=1}^{m} Y_j\right) \ = \ \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{C}ov(X_i, Y_j)$$

$$\mathbb{V}ar\left(\sum_{i=1}^{n} X_i\right) \ = \ \sum_{i=1}^{n} \mathbb{V}ar(X_i) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathbb{C}ov(X_i, X_j)$$

Se $X_1, \ldots, X_n$ são independentes, então: $\mathbb{V}ar(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \mathbb{V}ar(X_i)$.

Desigualdades de Markov e Chebyshev

Markov: v.a. $X \geq 0$, $a > 0$, $\alpha > 1$:

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}(X)}{a} \qquad ou \qquad \mathbb{P}(X \geq \alpha \cdot \mathbb{E}(X)) \leq \frac{1}{\alpha}$$

Prova:

- $\mathbb{E}(X) = \sum_{k=0}^{\infty} k \cdot \mathbb{P}(X = k) \geq \sum_{k=a}^{\infty} k \cdot \mathbb{P}(X = k) \geq$
- $\mathbb{E}(X) \geq a \cdot \sum_{k=a}^{\infty} \mathbb{P}(X = k) = a \cdot \mathbb{P}(X \geq a)$ $\square$

Chebyshev: v.a. X, $b > 0$, $\beta > 1$:

$$\mathbb{P}\left(\left|X - \mathbb{E}(X)\right| \geq b\right) \leq \frac{\mathbb{V}ar(X)}{b^2} \qquad ou \qquad \mathbb{P}\left(\left|X - \mathbb{E}(X)\right| \geq \beta \cdot \sigma(X)\right) \leq \frac{1}{\beta^2}$$

Prova:

$$\mathbb{P}\left(\left|X - \mathbb{E}(X)\right| \geq b\right) = \mathbb{P}\left((X - \mathbb{E}(X))^2 \geq b^2\right) \leq \frac{\mathbb{E}((X - \mathbb{E}(X))^2)}{b^2} = \frac{\mathbb{V}ar(X)}{b^2}$$

Distribuições Discretas: Bernoulli(p) e Binomial(n, p)

$X \sim Bernoulli(p)$: **1 (sucesso) com prob p, e 0 com prob $1-p$.**

$\mathbb{P}(X=1) = p$ e $\mathbb{P}(X=0) = 1-p$. $\quad \mathbb{E}(X) = 1 \cdot p + 0 \cdot (1-p) = p$.

$\mathbb{V}ar(X) = \mathbb{E}(X^2) - \mathbb{E}^2(X) = \mathbb{E}(X) - p^2 = p - p^2 = p(1-p)$.

$X \sim Binomial(n,p) = \sum_{i=1}^{n} Bernoulli(p)$: **número de sucessos (de probabilidade p) em n experimentos independentes.**

$$\mathbb{P}(X=k) = \binom{n}{k} p^k (1-p)^{n-k} \quad \Rightarrow \quad \sum_{k=0}^{n} \mathbb{P}(X=k) = (p+1-p)^n = 1.$$

$$\mathbb{E}(X) = \sum_{k=0}^{n} k \cdot \mathbb{P}(X=k) = \sum_{k=0}^{n} k \binom{n}{k} p^k (1-p)^{n-k} = n \cdot p,$$

pois $X = \sum_{i=1}^{n} Bernoulli(p)$ e portanto $\mathbb{E}(X) = \sum_{i=1}^{n} \mathbb{E}(X_i) = n \cdot p$ e

$\mathbb{V}ar(X) = \sum_{i=1}^{n} \mathbb{V}ar(X_i) = n \cdot p \cdot (1-p)$ (pois os X_i's são independentes entre si).

Distribuições Discretas: Bernoulli(p) e Binomial(n, p)

Aplicação: n moedas não-viciadas lançadas. Seja X o número de caras.
$X \sim Binomial(n, \frac{1}{2})$, $\mathbb{E}(X) = n/2$ e $\mathbb{V}ar(X) = n \cdot \frac{1}{2}(1 - \frac{1}{2}) = n/4$.

Markov: $\mathbb{P}(X \geq 3n/4) \leq \frac{\mathbb{E}(X)}{3n/4} = \frac{n/2}{3n/4} = \frac{2}{3}$ **(ruim)**

Chebyshev: $\mathbb{P}(X \geq 1.5\frac{n}{2}) \leq \mathbb{P}(|X - \mathbb{E}(X)| \geq 0.5\frac{n}{2}) \leq \frac{\mathbb{V}ar(X)}{(n/4)^2} = \frac{4}{n}$. **(bom)**

Chebyshev: $\mathbb{P}(X \geq 1.1\frac{n}{2}) \leq \mathbb{P}(|X - \mathbb{E}(X)| \geq 0.1\frac{n}{2}) \leq \frac{\mathbb{V}ar(X)}{(0.1 \cdot n/2)^2} = \frac{100}{n}$.

Chebyshev: $\mathbb{P}(X \geq 1.01\frac{n}{2}) \leq \mathbb{P}(|X - \mathbb{E}(X)| \geq 0.01\frac{n}{2}) \leq \frac{\mathbb{V}ar(X)}{(0.01 \cdot n/2)^2} = \frac{10000}{n}$.

Chebyshev: $\mathbb{P}(X \geq \frac{n+\sqrt{n}}{2}) \leq \mathbb{P}(|X - \mathbb{E}(X)| \geq \frac{\sqrt{n}}{2}) \leq \frac{\mathbb{V}ar(X)}{(\sqrt{n}/2)^2} = 1$. **(ruim)**

Nas próximas aulas, veremos outras distribuições clássicas e desigualdades mais fortes, como a de Chernoff.

Método Probabilístico: subgrafo bipartido grande

Teorema: Todo grafo $G = (V, E)$ possui um subgrafo bipartido com pelo menos $m/2$ (metade) das arestas, onde $m = |E|$ e $n = |V|$.
Prova:

- Seja A um subconjunto aleatório de vértices: $\mathbb{P}(v \in A) = p = 1/2$.
- Seja $B = V - A$ e seja X a v.a. do num arestas entre A e B.
- $X = \sum_{uv \in E} X_{uv}$, onde $X_{uv} = 1$, se uv está entre A e B, e 0 c.c.
- $\mathbb{E}(X_{uv}) = 1 \cdot (p \cdot (1-p) + (1-p) \cdot p) = \frac{1}{2}$
- $\mathbb{E}(X) = m/2$. Logo existe subgrafo bipartido com $m/2$ arestas. $\square$

Algoritmo Probabilístico: Seja A inicialmente vazio. Para cada vértice de G, coloque-o em A com probabilidade $1/2$. Seja $B = V - A$. Remova todas as arestas entre vértices de A e remova todas as arestas entre vértices de B. O subgrafo resultante é bipartido com número esperado de arestas igual a $m/2$.

X_{uv} são v.a.'s *Bernoulli*$(\frac{1}{2})$. Vamos mostrar que são independentes.
X_{uv} e X_{wz} são independentes, pois
$\mathbb{P}(X_{u,v} = 1, X_{w,z} = 1) = (\frac{1}{2}) \cdot (\frac{1}{2}) = \frac{1}{4} = \mathbb{P}(X_{u,v} = 1) \cdot \mathbb{P}(X_{w,z} = 1)$.
X_{uv} e X_{uz} são independentes, pois (para $p \neq 1/2$, não são independentes)
$\mathbb{P}(X_{u,v} = 1, X_{u,z} = 1) = (\frac{1}{2}) \cdot (\frac{1}{2}) = \frac{1}{4} = \mathbb{P}(X_{u,v} = 1) \cdot \mathbb{P}(X_{u,z} = 1)$.

Método Probabilístico: subgrafo bipartido grande

Teorema: Todo grafo $G = (V, E)$ possui um subgrafo bipartido com pelo menos $m/2$ (metade) das arestas, onde $m = |E|$ e $n = |V|$.
Prova:

- Seja A um subconjunto aleatório de vértices: $\mathbb{P}(v \in A) = p = 1/2$.
- Seja $B = V - A$ e seja X a v.a. do num arestas entre A e B.
- $X = \sum_{uv \in E} X_{uv}$, onde $X_{uv} = 1$, se uv está entre A e B, e 0 c.c.
- $\mathbb{E}(X_{uv}) = 1 \cdot (p \cdot (1-p) + (1-p) \cdot p) = \frac{1}{2}$
- $\mathbb{E}(X) = m/2$. Logo existe subgrafo bipartido com $m/2$ arestas. $\square$

Algoritmo Probabilístico: Seja A inicialmente vazio. Para cada vértice de G, coloque-o em A com probabilidade $1/2$. Seja $B = V - A$. Remova todas as arestas entre vértices de A e remova todas as arestas entre vértices de B. O subgrafo resultante é bipartido com número esperado de arestas igual a $m/2$.

$X = \sum_{uv \in E} X_{uv}$ é v.a. $X \sim Binomial(m, \frac{1}{2}) \quad \Rightarrow \quad \mathbb{V}ar(X) = \frac{m}{4}$

Chebyshev: $\mathbb{P}(X \leq 0.9\frac{m}{2}) \leq \mathbb{P}(|X - \mathbb{E}(X)| \geq 0.1\frac{m}{2}) \leq \frac{\mathbb{V}ar(X)}{(0.1 \cdot m/2)^2} = \frac{100}{m}$.

Nas próximas aulas, veremos outras distribuições clássicas e desigualdades mais fortes, como a de Chernoff.

Distribuições Discretas: Geométrica(p)

$X \sim Geom(p)$: **Num experimentos indep até 1 sucesso (de prob p).**

$\mathbb{P}(X = n) = p(1-p)^{n-1}$. $\sum_{n=1}^{\infty} \mathbb{P}(X = n) = \frac{p}{1-(1-p)} = 1$.

$$\sum_{n=1}^{\infty} n \cdot x^{n-1} = \frac{d}{dx}\left(\sum_{n=1}^{\infty} x^n\right) = \frac{d}{dx}\left(\frac{1}{1-x}\right) = \frac{1}{(1-x)^2} \qquad \text{para } 0 < x < 1$$

$$\sum_{n=1}^{\infty} n(n-1) \cdot x^{n-2} = \frac{d}{dx}\left(\sum_{n=1}^{\infty} n \cdot x^{n-1}\right) = \frac{d}{dx}\left(\frac{1}{(1-x)^2}\right) = \frac{2}{(1-x)^3}$$

$$\mathbb{E}(X) = \sum_{n=1}^{\infty} n \cdot \mathbb{P}(X = n) = \sum_{n=1}^{\infty} n \cdot p(1-p)^{n-1} = p \cdot \frac{1}{(1-(1-p))^2} = \frac{1}{p}$$

$$\mathbb{E}(X^2) = \sum_{n=1}^{\infty} n^2 \cdot \mathbb{P}(X = n) = \frac{2p(1-p)}{p^3} + \frac{p(1-p)}{(1-p)p^2} = \frac{2-p}{p^2}$$

$$\mathbb{V}ar(X) = \mathbb{E}(X^2) - \mathbb{E}^2(X) = \frac{2-p}{p^2} - \frac{1}{p^2} = \frac{1-p}{p^2}$$

Distribuições Discretas: Binomial Negativa(k, p)

$X \sim BN(k, p)$: **Num experimentos indep até k sucessos (de prob p).**
$\mathbb{P}(X = n) = \binom{n-1}{k-1} p^k (1-p)^{n-k}$.

v.a. $X \sim BN(k, p)$ pode ser vista como $X = \sum_{i=1}^{k} X_i$: a soma de k v.a. $X_i \sim Geom(p)$. Ou seja, $BN(k, p) = \sum_{i=1}^{k} Geom(p)$.

$$\mathbb{E}(X) = \sum_{i=1}^{k} \mathbb{E}(X_i) \;=\; k \cdot \frac{1}{p}$$

$$\mathbb{V}ar(X) = \sum_{i=1}^{k} \mathbb{V}ar(X_i) \;=\; k \cdot \frac{1-p}{p^2}$$

Problema do Colecionador de Cupons/Figurinhas

Álbum com n diferentes tipos de figurinhas. Cada figurinha comprada tem probabilidade igual para ser de qualquer tipo. Determinar esperança e variância do número de figurinhas compradas para encher o álbum.

Seja v.a. X esse número. Seja X_i o número a comprar até obter uma nova figurinha, dado que já possui i figurinhas.

$X = \sum_{i=0}^{n-1} X_i$, onde $X_i \sim Geom(\frac{n-i}{n})$. X é semelhante à Binomial Negativa.

$$\mathbb{E}(X) = \sum_{i=0}^{n-1} \mathbb{E}(X_i) \; = \; \sum_{i=0}^{n-1} \frac{n}{n-i} = n \cdot \sum_{k=1}^{n} \frac{1}{k} \approx n \ln n + 0.5772 \cdot n + \frac{1}{2},$$

pois $\sum_{k=1}^{n} \frac{1}{k} = \ln n + 0,5772 + \frac{1}{2n} + O(\frac{1}{n^2})$. Ademais, $0 \leq \sum_{k=1}^{n} \frac{1}{k} - \ln n \leq 1$.

$$\mathbb{V}ar(X) = \sum_{i=0}^{n-1} \mathbb{V}ar(X_i) \; = \; n \sum_{i=0}^{n-1} \frac{i}{(n-i)^2} = n \sum_{k=1}^{n} \frac{n-k}{k^2} = n^2 \sum_{k=1}^{n} \frac{1}{k^2} - n \sum_{k=1}^{n} \frac{1}{k}$$

$$\mathbb{V}ar(X) = n^2 \sum_{k=1}^{n} \frac{1}{k^2} - n \sum_{k=1}^{n} \frac{1}{k} \approx n^2 \frac{\pi^2}{6} - n \ln n - 0,5772 \cdot n - \frac{1}{2},$$

pois $\lim_{n \to \infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$.

Problema do Colecionador de Cupons/Figurinhas

Álbum com n diferentes tipos de figurinhas. Cada figurinha comprada tem probabilidade igual para ser de qualquer tipo. Determinar esperança e variância do número X de figurinhas compradas para encher o álbum.

$$\mathbb{E}(X) = \sum_{i=0}^{n-1} \mathbb{E}(X_i) \;=\; \sum_{i=0}^{n-1} \frac{n}{n-i} = n \cdot \sum_{k=1}^{n} \frac{1}{k} \;\approx\; n \ln n \;+\; 0,5772 \cdot n \;+\; \frac{1}{2}.$$

$$\mathbb{V}ar(X) = n^2 \sum_{k=1}^{n} \frac{1}{k^2} - n \sum_{k=1}^{n} \frac{1}{k} \;\approx\; n^2 \cdot \frac{\pi^2}{6} - n \cdot \ln n - 0,5772 \cdot n - \frac{1}{2}.$$

Exemplo: Álbum da copa: $n = 680$ figurinhas.

$\mathbb{E}(X) = 4828.03 \approx 4828.02, \qquad \sigma(X) = \sqrt{\mathbb{V}ar(X)} = 869.0 \approx 869.4.$

Exercício: (a) Implementar simulação do problema do colecionador de figurinhas para $n = 680$, repetir 5000 vezes obtendo o número de figurinhas compradas até preencher o álbum, calcular média e desvio padrão. (b) Fazer o mesmo considerando a venda em pacotes de 5 figurinhas diferentes entre si. (c) Fazer gráficos das distribuições dos itens (a) e (b) obtidas nas simulações.

Problema do Colecionador de Cupons/Figurinhas - 2

Álbum com n diferentes tipos de figurinhas. Comprando k figurinhas com álbum vazio, determinar esperança e variância do número de figurinhas no álbum.

Seja v.a. $X = \sum_{i=1}^{n} X_i$ esse número, onde $X_i = 1$, se a figurinha i foi comprada, e 0 c.c.

$\mathbb{P}(X_i = 0) = \left(1 - \frac{1}{n}\right)^k$ e $\mathbb{P}(X_i = 1) = 1 - \left(\frac{n-1}{n}\right)^k$

$X_i \sim Bernoulli\left(1 - \left(\frac{n-1}{n}\right)^k\right)$ (não independentes entre si)

$\mathbb{E}(X_i) = 1 - \left(\frac{n-1}{n}\right)^k$ e $\mathbb{V}ar(X_i) = \left(\frac{n-1}{n}\right)^k \cdot \left(1 - \left(\frac{n-1}{n}\right)^k\right)$

$\mathbb{E}(X) = n \cdot \left(1 - \left(\frac{n-1}{n}\right)^k\right)$. Exemplo: $n = k = 680 \Rightarrow \mathbb{E}(X) = 430$

Para $k = c \cdot n$, temos: $\mathbb{E}(X) = n \cdot \left(1 - \left(1 - \frac{1}{n}\right)^{n \cdot c}\right) \approx n \cdot \left(1 - \frac{1}{e^c}\right)$,

pois $lim_{n \to \infty}(1 + \frac{\alpha}{n})^n = e^{\alpha}$. Exemplo: $n = k = 680 \Rightarrow \mathbb{E}(X) = 430$

Problema do Colecionador de Cupons/Figurinhas - 2

Álbum com n diferentes tipos de figurinhas. Comprando k figurinhas com álbum vazio, determinar esperança e variância do número de figurinhas no álbum.

v.a. $X = \sum_{i=1}^{n} X_i$ esse número, onde $X_i = 1$, se a figurinha i foi comprada, e 0 c.c.

$\mathbb{P}(X_i = 0) = \left(1 - \frac{1}{n}\right)^k$ $\qquad$ $X_i \sim Bernoulli\left(1 - \left(\frac{n-1}{n}\right)^k\right)$

$\mathbb{E}(X) = n \cdot \left(1 - \left(\frac{n-1}{n}\right)^k\right)$. $\qquad$ Exemplo: $n = k = 680 \Rightarrow \mathbb{E}(X) = 430$

$$\mathbb{V}ar(X) = \mathbb{V}ar\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \mathbb{V}ar(X_i) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathbb{C}ov(X_i, X_j)$$

$\mathbb{C}ov(X_i, X_j) = \mathbb{E}(X_i X_j) - \mathbb{E}(X_i)\mathbb{E}(X_j)$. $\qquad$ $\mathbb{E}(X_i X_j) = \mathbb{P}(X_i X_j = 1) = \mathbb{P}(A_i \cap A_j)$

$\mathbb{P}(A_i \cap A_j) = 1 - \mathbb{P}(\overline{A_i \cap A_j}) = 1 - \mathbb{P}(\overline{A_i} \cup \overline{A_j}) = 1 - \mathbb{P}(\overline{A_i}) - \mathbb{P}(\overline{A_j}) + \mathbb{P}(\overline{A_i} \cap \overline{A_j})$

$\mathbb{E}(X_i X_j) = 1 - 2\left(\frac{n-1}{n}\right)^k + \left(\frac{n-2}{n}\right)^k$

$$\mathbb{C}ov(X_i, X_j) = \left(\frac{n-2}{n}\right)^k - \left(\frac{n-1}{n}\right)^{2k}$$

Problema do Colecionador de Cupons/Figurinhas - 2

Álbum com n diferentes tipos de figurinhas. Comprando k figurinhas com álbum vazio, determinar esperança e variância do número de figurinhas no álbum.

Seja v.a. $X = \sum_{i=1}^{n} X_i$ o número, onde $X_i = 1$, se a figurinha i foi comprada, e 0 c.c.

$\mathbb{P}(X_i = 0) = \left(1 - \frac{1}{n}\right)^k$ e $\mathbb{P}(X_i = 1) = 1 - \left(\frac{n-1}{n}\right)^k$

$X_i \sim Bernoulli\left(1 - \left(\frac{n-1}{n}\right)^k\right)$ (não independentes entre si)

$\mathbb{E}(X_i) = 1 - \left(\frac{n-1}{n}\right)^k$ e $\mathbb{V}ar(X_i) = \left(\frac{n-1}{n}\right)^k \cdot \left(1 - \left(\frac{n-1}{n}\right)^k\right)$

$\mathbb{E}(X) = n \cdot \left(1 - \left(\frac{n-1}{n}\right)^k\right)$. Exemplo: $n = k = 680 \Rightarrow \mathbb{E}(X) = 430$

$$\mathbb{V}ar(X) = \mathbb{V}ar\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \mathbb{V}ar(X_i) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \mathbb{C}ov(X_i, X_j)$$

$$\mathbb{C}ov(X_i, X_j) = \left(\frac{n-2}{n}\right)^k - \left(\frac{n-1}{n}\right)^{2k}$$

$$\mathbb{V}ar(X) = n\left(\frac{n-1}{n}\right)^k \left(1 - \left(\frac{n-1}{n}\right)^k\right) + n(n-1)\left(\left(\frac{n-2}{n}\right)^k - \left(\frac{n-1}{n}\right)^{2k}\right)$$

$$\mathbb{V}ar(X) = n\left(\frac{n-1}{n}\right)^k + n(n-1)\left(\frac{n-2}{n}\right)^k - n^2\left(\frac{n-1}{n}\right)^{2k}$$

Problema do Colecionador de Cupons/Figurinhas - 2

Álbum com n diferentes tipos de figurinhas. Comprando k figurinhas com álbum vazio, determinar esperança e variância do número de figurinhas no álbum.

Seja v.a. $X = \sum_{i=1}^{n} X_i$ o número, onde $X_i = 1$, se a figurinha i foi comprada, e 0 c.c.

$\mathbb{P}(X_i = 0) = \left(1 - \frac{1}{n}\right)^k$ e $\mathbb{P}(X_i = 1) = 1 - \left(\frac{n-1}{n}\right)^k$

$X_i \sim Bernoulli\left(1 - \left(\frac{n-1}{n}\right)^k\right)$ (não independentes entre si)

$\mathbb{E}(X_i) = 1 - \left(\frac{n-1}{n}\right)^k$ e $\mathbb{V}ar(X_i) = \left(\frac{n-1}{n}\right)^k \cdot \left(1 - \left(\frac{n-1}{n}\right)^k\right)$

$$\mathbb{E}(X) = n \cdot \left(1 - \left(\frac{n-1}{n}\right)^k\right) \approx n \cdot \left(1 - \frac{1}{e}\right) \quad \text{(para } k = n \text{ grande)}.$$

$$\mathbb{V}ar(X) = n\left(\frac{n-1}{n}\right)^k + n(n-1)\left(\frac{n-2}{n}\right)^k - n^2\left(\frac{n-1}{n}\right)^{2k} \approx n \cdot \left(\frac{1}{e} - \frac{2}{e^2}\right)$$

Exemplo: $n = k = 680$: $\mathbb{E}(X) = 430$, $\mathbb{V}ar(X) = 66.1$, $\sigma(X) = \sqrt{66.1} = 8.13$

Exercício: (a) Implementar simulação do problema do colecionador de figurinhas-2 para $n = k = 680$, repetir 5000 vezes obtendo o número de figurinhas novas no álbum, calcular média e desvio padrão. (b) Fazer o mesmo considerando a venda em pacotes de 5 figurinhas diferentes entre si. (c) Fazer gráficos das distribuições dos itens (a) e (b) obtidas nas simulações.

Distribuições Discretas: HiperGeométrica(N, K, n)

$X \sim HiperGeom(N, K, n)$: **Número de sucessos em n retiradas (sem reposição) de uma população de tamanho N com K sucessos.**

$$\mathbb{P}(X = k) = \frac{\binom{K}{k} \cdot \binom{N-K}{n-k}}{\binom{N}{n}}. \qquad \sum_{k=0}^{K} \mathbb{P}(X = k) = \sum_{k=0}^{K} \frac{\binom{K}{k} \cdot \binom{N-K}{n-k}}{\binom{N}{n}} = 1 \text{ (Vandermonde)}$$

$$\mathbb{E}(X) = \sum_{k=0}^{K} k \cdot \mathbb{P}(X = k) = \frac{nK}{N} \sum_{k=1}^{K} \frac{\binom{K-1}{k-1}\binom{N-K}{n-k}}{\binom{N-1}{n-1}} = \frac{nK}{N} \sum_{\ell=0}^{K-1} \frac{\binom{K-1}{\ell}\binom{N-1-(K-1)}{n-1-\ell}}{\binom{N-1}{n-1}} = \frac{nK}{N}$$

$$\mathbb{E}(X^2) = \sum_{k=0}^{K} k^2 \cdot \mathbb{P}(X = k) = \ldots \text{ contas } \ldots = \frac{nK(n-1)(K-1)}{N(N-1)} + \frac{nK}{N}$$

$$\mathbb{V}ar(X) = \mathbb{E}(X^2) - \mathbb{E}^2(X) = \frac{nK(n-1)(K-1)}{N(N-1)} + \frac{nK}{N} - \left(\frac{nK}{N}\right)^2 = \frac{nK(N-K)}{N^2} \cdot \left(\frac{N-n}{N-1}\right)$$

Distribuições Discretas: Poisson(λ)

$X \sim Poisson(\lambda)$: **Aproximação útil p/** $Binomial(n,p)$ **quando** $\lambda = np$ **tende a uma constante para** n **muito grande (e** p **muito pequeno).** Também mede número de sucessos. Exemplo: Número de meteoritos com mais de 1m de diâmetro que atingem a Terra por ano.

$$\mathbb{P}(X=k) = \frac{e^{-\lambda}\cdot\lambda^k}{k!}. \qquad \sum_{k=0}^{\infty}\mathbb{P}(X=k) = \sum_{k=0}^{K}\frac{e^{-\lambda}\lambda^k}{k!} = 1 \ \text{(Taylor p/ exp)}$$

$$\mathbb{E}(X) = \sum_{k=0}^{\infty} k\cdot\mathbb{P}(X=k) = \lambda\cdot\sum_{k=1}^{\infty}\frac{e^{-\lambda}\cdot\lambda^{k-1}}{(k-1)!} = \lambda\cdot\sum_{\ell=0}^{\infty}\frac{e^{-\lambda}\cdot\lambda^{\ell}}{\ell!} = \lambda$$

$$\mathbb{E}(X^2) = \sum_{k=0}^{\infty} k^2\cdot\mathbb{P}(X=k) = \lambda\cdot\sum_{k=1}^{\infty} k\frac{e^{-\lambda}\cdot\lambda^{k-1}}{(k-1)!} = \text{...contas...} = \lambda^2+\lambda$$

$$\mathbb{V}ar(X) = \mathbb{E}(X^2) - \mathbb{E}^2(X) = (\lambda^2+\lambda) - \lambda^2 = \lambda$$

Aproximação: Considere $n \to \infty$, $p \to 0$ e $np \to \lambda$ constante.

$$\mathbb{P}(Bin(n,p)=k) = \binom{n}{k}p^k(1-p)^{n-k} = \frac{n!}{k!(n-k)!}p^k(1-p)^{n-k} \to \frac{n^k}{k!}\cdot p^k(1-p)^n$$

$$\mathbb{P}(Bin(n,p)=k) \to \frac{\lambda^k}{k!}\left(1-\frac{\lambda}{n}\right)^n \to \frac{e^{-\lambda}\cdot\lambda^k}{k!} = \mathbb{P}(Poisson(\lambda)=k)$$

Paradigma de Poisson

Poisson é aproximação útil para Binomial mesmo se experimentos de Bernoulli têm probabilidades diferentes p_i (mas pequenas) e são fracamente dependentes. Nesse caso, $\lambda = \sum_{i=1}^{n} p_i$.

Exemplo: Menor n para que seja mais provável encontrar 2 pessoas com mesmo aniversário do que o contrário.

Solução 1: Probabilidade de todos diferentes:
$\prod_{k=1}^{n-1} \left(1 - \frac{k}{365}\right) \leq \frac{1}{2} \;\Rightarrow\; n \geq 23.$

Solução 2: Poisson: v.a. $X_{i,j} = 1$ se pessoas i e j tem mesmo aniversário. $X_{i,j}$ são fracamente dependentes. São $\binom{n}{2}$ experimentos de Bernoulli com probabilidade $\frac{1}{365}$. **Assim:** $\lambda = \binom{n}{2} \cdot \frac{1}{365}$.

$$\mathbb{P}\left(\sum X_{i,j} = 0\right) \approx \mathbb{P}(Poisson(\lambda) = 0) = \frac{e^{-\lambda} \cdot \lambda^0}{0!} \leq \frac{1}{2} \Rightarrow n(n-1) \geq 730 \ln 2 = 506$$

Portanto $n \geq 23$.

Paradigma de Poisson

Poisson é aproximação útil para Binomial mesmo se experimentos de Bernoulli têm probabilidades diferentes p_i (mas pequenas) e são fracamente dependentes. Nesse caso, $\lambda = \sum_{i=1}^{n} p_i$.

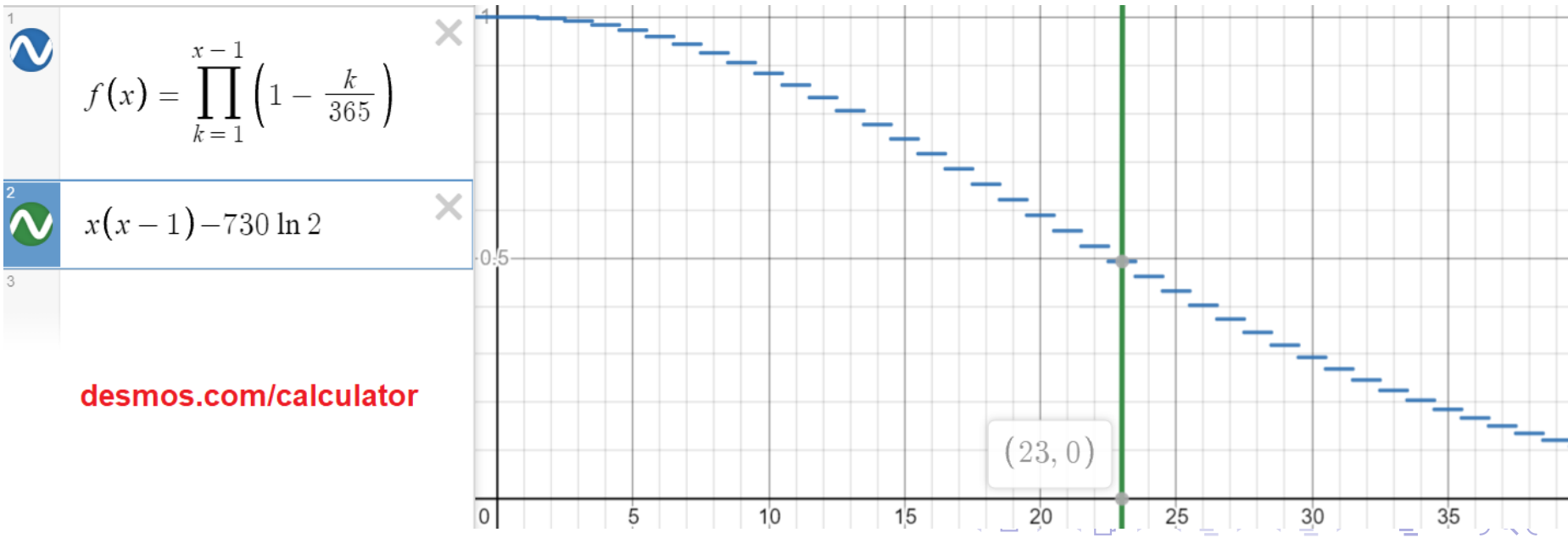
Exemplo: Menor n para que seja mais provável encontrar 2 pessoas com mesmo aniversário do que o contrário.

Solução 1: Probabilidade de todos diferentes: $\prod_{k=1}^{n-1} \left(1 - \frac{k}{365}\right) \leq \frac{1}{2} \Rightarrow n \geq 23$.

Solução 2: Poisson: v.a. $X_{i,j} = 1$ se pessoas i e j tem mesmo aniversário. $X_{i,j}$ são fracamente dependentes. São $\binom{n}{2}$ experimentos de Bernoulli com probabilidade $\frac{1}{365}$. **Assim:** $\lambda = \binom{n}{2} \cdot \frac{1}{365}$.

$$\mathbb{P}\left(\sum X_{i,j} = 0\right) \approx \mathbb{P}(Poisson(\lambda) = 0) = \frac{e^{-\lambda} \cdot \lambda^0}{0!} \leq \frac{1}{2} \Rightarrow n(n-1) \geq 730 \ln 2 = 506$$

Portanto $n \geq 23$.



Distribuições Discretas: Uniforme$\{a, b\}$

$X \sim Unif\{a, b\}$ p/ $a, b \in \mathbb{Z}$: **X é número inteiro entre a e b, inclusive, todos com a mesma probabilidade** $\mathbb{P}(X = k) = \frac{1}{b-a+1}$**.**

$$\mathbb{E}(X) = \sum_{k=a}^{b} k \cdot \mathbb{P}(X = k) \;=\; \frac{(b+a)(b-a+1)}{2(b-a+1)} \;=\; \frac{b+a}{2}$$

$$\mathbb{E}(X^2) = \sum_{k=a}^{b} k^2 \cdot \mathbb{P}(X = k) \;=\; \frac{b(b+1)(2b+1)}{6(b-a+1)} - \frac{(a-1)a(2a-1)}{6(b-a+1)}$$

$$\mathbb{V}ar(X) = \mathbb{E}(X^2) - \mathbb{E}^2(X) \;=\; \ldots \;=\; \frac{(b-a)(b-a+2)}{12}$$

$$\sigma(X) = \sqrt{\mathbb{V}ar(X)} \;\approx\; \frac{(b-a)}{2\sqrt{3}}, \qquad \text{para } b-a \text{ muito grande.}$$

Distribuições Contínuas: Uniforme$[a, b]$

$X \sim Unif[a, b]$ p/ $a, b \in \mathbb{R}$: X **é número real entre** a **e** b**, inclusive, todos com a mesma probabilidade.**

Função densidade de prob $f(x) = \frac{1}{b-a}$ se $a \leq x \leq b$; $f(x) = 0$, cc.

$$\mathbb{E}(X) = \int_a^b f(x) \cdot x dx = \frac{x^2}{2(b-a)}\Big|_a^b = \frac{b^2 - a^2}{2(b-a)} = \frac{b+a}{2}$$

$$\mathbb{E}(X^2) = \int_a^b f(x) \cdot x^2 dx = \frac{x^3}{3(b-a)}\Big|_a^b = \frac{b^3 - a^3}{3(b-a)} = \frac{b^2 + ab + a^2}{3}$$

$$\mathbb{V}ar(X) = \mathbb{E}(X^2) - \mathbb{E}^2(X) = \frac{b^2 + ab + a^2}{3} - \left(\frac{b+a}{2}\right)^2 = \frac{(b-a)^2}{12}$$

$$\sigma(X) = \sqrt{\mathbb{V}ar(X)} = \frac{(b-a)}{2\sqrt{3}}$$

Distribuições Contínuas: Normal(μ, σ^2)

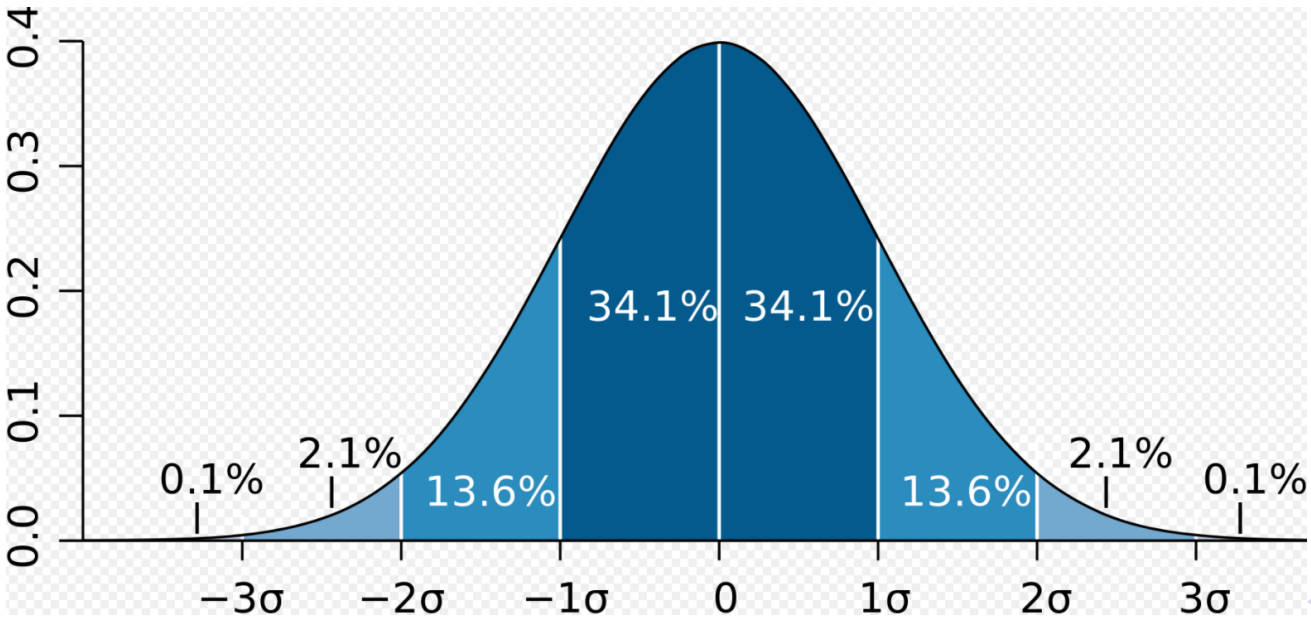
$X \sim Normal(\mu, \sigma^2)$: X **é v.a. real com média** μ **e variância** σ^2

Função densidade de prob $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$.

$$\mathbb{E}(X) = \int_a^b f(x) \cdot x dx \;=\; \ldots \;=\; \mu$$

$$\mathbb{E}(X^2) = \int_a^b f(x) \cdot x^2 dx \;=\; \ldots \;=\; \sigma^2 + \mu^2$$

$$\mathbb{V}ar(X) = \mathbb{E}(X^2) - \mathbb{E}^2(X) \;=\; \sigma^2. \qquad \sigma(X) = \sqrt{\mathbb{V}ar(X)} \;=\; \sigma$$



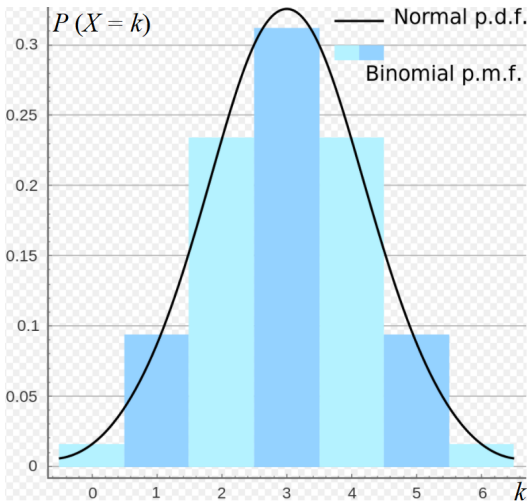
Distribuição Binomial tende a Normal para $n \to \infty$

Seja $X \sim Binomial(n, p)$. Por Stirling: $n! \approx n^n \cdot e^{-n} \cdot \sqrt{2\pi n}$:

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \approx \frac{n^n e^{-n} \sqrt{2\pi n} \cdot p^k (1-p)^{n-k}}{k^k e^{-k} \sqrt{2\pi k} \cdot (n-k)^{n-k} e^{-(n-k)} \sqrt{2\pi (n-k)}}$$

$$= \left(\frac{np}{k}\right)^k \left(\frac{n(1-p)}{n-k}\right)^{n-k} \sqrt{\frac{n}{2\pi k(n-k)}} = \ldots =$$

$$= \frac{1}{\sqrt{2\pi np(1-p)}} \cdot \exp\left\{-\frac{(k-np)^2}{2np(1-p)}\right\} = \mathbb{P}\Big(Normal(np, np(1-p)) = k\Big)$$



Geração de v.a. Uniforme$(0, 1)$ no computador

Alerta 1: computador não gera números reais. Aproximação razoável usando 64 bits: divide o intervalo $[0, 1]$ em 2^{64} subintervalos.

Alerta 2: computador é determinístico. Aproximação razoável: simular comportamento probabilístico com gerador pseudo-aleatório. Os melhores geradores passam em testes estatísticos para medir a independência das amostras geradas e sua aleatoriedade.

Suposição: Temos algoritmo *randUni* que gera uma v.a. Uniforme$(0, 1)$.

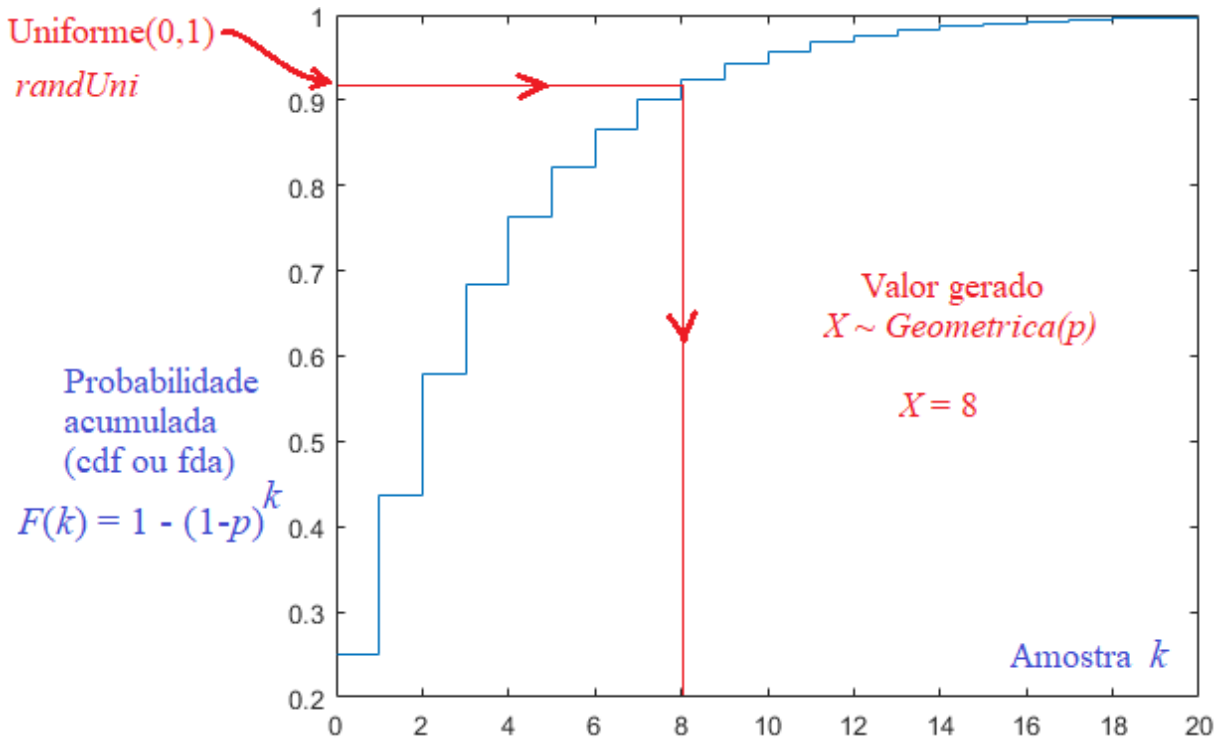
$X \sim Bernoulli(p)$: Se $(randUni < p)$, $X = 1$; caso contrário, $X = 0$.

$X \sim Uniforme\{a, b\}$: $X = a + \lfloor randUni \cdot (b - a + 1) \rfloor$.

Geração de v.a. $X \sim Geometrica(p)$ no computador

Algoritmo 1: Gera v.a.'s $Bernoulli(p)$ até obter um sucesso (1).

Algoritmo 2: Gera $Uniforme(0,1)$ e aplica a inversa da função de probabilidade acumulada.



$$F(k) = \mathbb{P}(X \leq k) = 1-(1-p)^k \;=\; u \;\Rightarrow\; F^{-1}(u) = k = \left\lceil \frac{\log(1-u)}{\log(1-p)} \right\rceil .$$

Geração de v.a. $X \sim Binomial(n, p)$ no computador

Geração de v.a. $X \sim Binomial(n, p)$:

Algoritmo 1: Gera n v.a.'s $Bernoulli(p)$ e conta o número de sucessos.

Algoritmo 2: Gera v.a.'s $Geometrica(p)$ até sua soma ultrapassar n e retorna o número de v.a.'s menos 1. Número esperado de geométricas geradas: $n \cdot p$.

Exemplo: $n = 10$, $p = 0.5$. $Geometrica(p)$ geradas: 2,3,1,2,4. $Binomial(n, p)$ gerada $= 4$.

Geração de v.a. $X \sim BinomialNegativa(k, p)$:

Algoritmo 1: Gera k v.a.'s $Geometrica(p)$ e soma seus valores.

Geração de Permutação aleatória no computador

Algoritmo 1: Gera n v.a.'s $Uniforme(0,1)$, ordena seus valores e substitui os valores pela sua posição no vetor ordenado. Tempo $O(n \log n)$.

Exemplo: $[0.6, 0.2, 0.7, 0.4, 0.3]$. Permutação = (4,1,5,3,2).

Algoritmo 2: Fisher-Yates Shuffle. Tempo $O(n)$

Algoritmo Fisher-Yates-Shuffle (inteiro n)

1 Cria vetor $perm[1, \ldots, n]$
2 **para** $k \leftarrow 1$ **até** n **faça**
3 $\quad perm[k] \leftarrow k$
4 **para** $k \leftarrow 1$ **até** n **faça**
5 $\quad \ell \leftarrow Uniforme\{k, n\}$
6 $\quad$ Trocar $perm[k] \leftrightarrow perm[\ell]$

Desigualdade de Chernoff (p/ v.a. Binomiais)

Chernoff: n v.a.'s **independentes** $X_1, \ldots, X_n \sim \mathit{Bernouli}(p)$
Seja $X = \sum_{i=1}^{n} X_i \sim \mathit{Binomial}(n, p)$ e $\mathbb{E}(X) = \sum_{i=1}^{n} \mathbb{E}(X_i) = np$:

$$\mathbb{P}\left(X > \mathbb{E}(X) + t \cdot n\right) \leq \exp\left\{-2t^2 \cdot n\right\}$$

$$\mathbb{P}\left(X < \mathbb{E}(X) - t \cdot n\right) \leq \exp\left\{-2t^2 \cdot n\right\}$$

Juntando os dois bounds em um só:

$$\mathbb{P}\left(|X - \mathbb{E}(X)| \geq t \cdot n\right) \leq 2 \cdot \exp\left\{-2t^2 \cdot n\right\}$$

$$\mathbb{P}\left(|X - \mathbb{E}(X)| \geq \alpha\right) \leq 2 \cdot \exp\left\{\frac{-2\alpha^2}{n}\right\}$$

$$\mathbb{P}\left(|X - \mathbb{E}(X)| \geq t \cdot \sqrt{n}\right) \leq 2 \cdot \exp\left\{-2t^2\right\}.$$

$$t = 3 \Rightarrow \mathit{prob} \approx 3 \cdot 10^{-8}$$

Desigualdade de Chernoff-Hoeffding (generalização)

Chernoff: n v.a.'s **independentes** $X_i \in [0,1]$ com média μ_i
As v.a.'s não precisam ser Bernoulli, nem ter a mesma distribuição.
Seja $X = \sum_{i=1}^{n} X_i$ e $\mathbb{E}(X) = \sum_{i=1}^{n} \mathbb{E}(X_i) = n \cdot \mu$, onde $\mu = (\mu_1 + \ldots + \mu_n)/n$

$$\mathbb{P}\left(X > \mathbb{E}(X) + t \cdot n\right) \leq \exp\left\{-2t^2 \cdot n\right\}$$

$$\mathbb{P}\left(X < \mathbb{E}(X) - t \cdot n\right) \leq \exp\left\{-2t^2 \cdot n\right\}$$

Juntando os dois bounds em um só:

$$\mathbb{P}\left(|X - \mathbb{E}(X)| \geq t \cdot n\right) \leq 2 \cdot \exp\left\{-2t^2 \cdot n\right\}$$

$$\mathbb{P}\left(|X - \mathbb{E}(X)| \geq \alpha\right) \leq 2 \cdot \exp\left\{\frac{-2\alpha^2}{n}\right\}$$

$$\mathbb{P}\left(|X - \mathbb{E}(X)| \geq t \cdot \sqrt{n}\right) \leq 2 \cdot \exp\left\{-2t^2\right\}.$$

$t = 3 \Rightarrow prob \approx 3 \cdot 10^{-8}$

Desigualdade de Chernoff-Hoeffding (generalização)

Chernoff: n v.a.'s **independentes** $X_i \in [a_i, b_i]$ com média μ_i
As v.a.'s não precisam ser Bernoulli, nem ter a mesma distribuição.
Seja $X = \sum_{i=1}^{n} X_i$ e $\mathbb{E}(X) = \sum_{i=1}^{n} \mathbb{E}(X_i) = n \cdot \mu$, onde $\mu = (\mu_1 + \ldots + \mu_n)/n$

$$\mathbb{P}(X > \mathbb{E}(X) + t \cdot n) \leq \exp\left\{ \frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \cdot n^2 \right\}$$

$$\mathbb{P}(X < \mathbb{E}(X) - t \cdot n) \leq \exp\left\{ \frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \cdot n^2 \right\}$$

Juntando os dois bounds em um só:

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq t \cdot n) \leq 2 \cdot \exp\left\{ \frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \cdot n^2 \right\}$$

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq \alpha) \leq 2 \cdot \exp\left\{ \frac{-2\alpha^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \right\}$$

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq t \cdot \sqrt{n}) \leq 2 \cdot \exp\left\{ \frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \cdot n \right\}$$

Desigualdade de Chernoff-Hoeffding (comparação)

Aplicação: n moedas não-viciadas lançadas. Seja X o número de caras.
$X \sim Binomial(n, \frac{1}{2})$, $\mathbb{E}(X) = n/2$ e $\mathbb{V}ar(X) = n \cdot \frac{1}{2}(1 - \frac{1}{2}) = n/4$.

Markov: $\mathbb{P}(X \geq 3n/4) \leq \frac{\mathbb{E}(X)}{3n/4} = \frac{n/2}{3n/4} = \frac{2}{3}$ **(ruim)**

Chebyshev: $\mathbb{P}(X \geq 1.5\frac{n}{2}) \leq \mathbb{P}(|X - \mathbb{E}(X)| \geq 0.5\frac{n}{2}) \leq \frac{\mathbb{V}ar(X)}{(n/4)^2} = \frac{4}{n}$. **(bom)**

Chebyshev: $\mathbb{P}(X \geq 1.01\frac{n}{2}) \leq \mathbb{P}(|X - \mathbb{E}(X)| \geq 0.01\frac{n}{2}) \leq \frac{\mathbb{V}ar(X)}{(0.01 \cdot n/2)^2} = \frac{10000}{n}$.

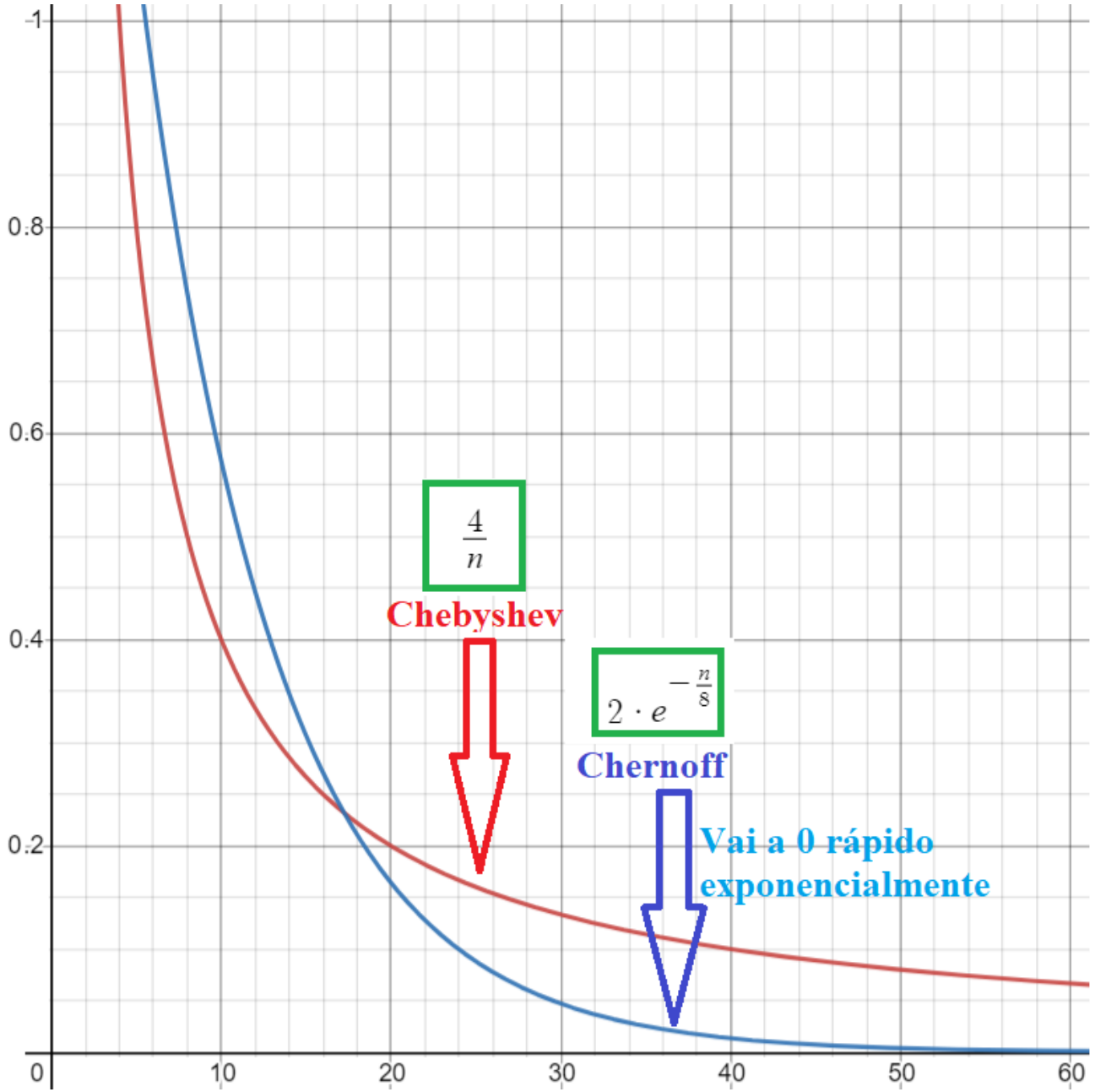
Chebyshev:
$\mathbb{P}(|X - \mathbb{E}(X)| \geq t\sqrt{n}) \leq \frac{\mathbb{V}ar(X)}{(t\sqrt{n})^2} = \frac{1}{4\ t^2}$. **(razoável)**

Chernoff:
$\mathbb{P}(|X - \mathbb{E}(X)| \geq \frac{1}{4} \cdot n) \leq 2e^{-2 \cdot (1/4)^2 \cdot n} = 2e^{-n/8}$. **(excelente)**

Chernoff:
$\mathbb{P}(|X - \mathbb{E}(X)| \geq \frac{0.01}{2} \cdot n) \leq 2e^{-2(0.01/2)^2 \cdot n} = 2e^{-n/20000}$. **(excelente)**

Chernoff:
$\mathbb{P}(|X - \mathbb{E}(X)| \geq t\sqrt{n}) \leq 2e^{-2t^2}$. **(excelente)**

Desigualdade de Chernoff-Hoeffding (comparação)



Chernoff: subgrafo bipartido grande

Problema: Dado um grafo com n vértices e m arestas, obter subgrafo bipartido com muitas arestas, próximo da metade $m/2$.

Algoritmo Probabilístico: Seja A inicialmente vazio. Para cada vértice de G, coloque-o em A com probabilidade $1/2$. Seja $B = V - A$. Remova todas as arestas entre vértices de A e remova todas as arestas entre vértices de B. O subgrafo resultante é bipartido com número esperado de arestas igual a $m/2$.

$X = \sum_{uv \in E} X_{uv}$ é v.a. $X \sim Binomial(m, \frac{1}{2}) \quad \Rightarrow \quad \mathbb{V}ar(X) = \frac{m}{4}$

Chebyshev: $\mathbb{P}(X \leq 0.9\frac{m}{2}) \leq \mathbb{P}(|X - \mathbb{E}(X)| \geq 0.1\frac{m}{2}) \leq \frac{\mathbb{V}ar(X)}{(0.1 \cdot m/2)^2} = \frac{100}{m}$.

Chernoff:
$\mathbb{P}(X \leq 0.9\frac{m}{2}) \leq \mathbb{P}(X \leq \mathbb{E}(X) - 0.1\frac{m}{2}) \leq e^{-2 \cdot (0.1/2)^2 m} = e^{-m/200}$.

Exemplo: $m = 1000$: Chebyshev 10%. Chernoff $< 0.7\%$
Exemplo: $m = 10000$: Chebyshev 1%. Chernoff $< 10^{-20}$

Chernoff: Amplificação para Classe BPP

Teorema: As seguintes afirmações são equivalentes, onde n é tam. da instância de Π:

(1) O Problema Π está em BPP (erro $\leq \frac{1}{3}$ *two sided error*)

(2) Existe um polinômio $q(n)$ tal que Π possui um algoritmo probabilístico polinomial com erro no máximo $\frac{1}{2} - \frac{1}{q(n)}$ (*two sided error*)

(3) Para todo polinômio $p(n) > 1$, Π possui um algoritmo probabilístico polinomial com erro no máximo $2^{-p(n)}$ (*two sided error*)

Proof: Claramente (3)⇒(1)⇒(2). Vamos provar que (2)⇒(3).

Seja M um algoritmo com erro $\leq \frac{1}{2} - \frac{1}{q(n)}$ que retorna SIM/NÃO. Seja M' um algoritmo que repete $t(n)$ vezes o algoritmo M sobre a mesma instância e retorna o valor SIM/NÃO que mais ocorre (majoritário), onde $t(n)$ será definido depois.

Seja X_i a v.a. indicadora que retorna 1 se a i-ésima execução de M retornou a resposta correta. Seja $X = \sum X_i$. Logo $\mathbb{E}(X) \geq t(\frac{1}{2} + \frac{1}{q})$. Usando Chernoff:

$$\mathbb{P}(X \leq t/2) \leq \mathbb{P}(X \leq \mathbb{E}(X) - t/q) \leq \exp\left\{-\frac{2(t/q)^2}{t}\right\} \leq \exp\left\{-\frac{2t}{q^2}\right\} \leq e^{-p(n)} \leq 2^{-p(n)},$$

para $t(n) = p(n) \cdot q^2(n)/2$, pois $\mathbb{E}(X) \leq t$.

Desigualdade de McDiarmid (Bounded Differences)

Função c-Lipschitz: Função $f(x_1, \ldots, x_n)$ tal que $|f(x) - f(x')| \leq c$ se x e x' diferem em só uma das n coordenadas.

Método das Diferenças Limitadas: *generaliza Chernoff-Hoeffding.* v.a. **independentes** $X_1, \ldots, X_n$ e função $f(x_1, \ldots, x_n)$ c-Lipschitz.

$$\mathbb{P}\left(\left|f(X_1, \ldots, X_n) - \mathbb{E}(f(X_1, \ldots, X_n))\right| > t \cdot n\right) \leq 2 \cdot \exp\left\{\frac{-2t^2}{c^2} \cdot n\right\}$$

Exemplo: Binomial. v.a. $X_1, X_2, \ldots, X_n \sim Bernoulli(p)$ e função $f(X_1, \ldots, X_n) = \sum_{i=1}^{n} X_i = X$.

$$\mathbb{P}\left(\left|X - \mathbb{E}(X)\right| > t \cdot n\right) \leq 2 \cdot \exp\left\{-2t^2 \cdot n\right\} \quad \text{(McDiarmid} \equiv \text{Chernoff)}$$

Exemplo: Soma dos senos. v.a. $X_1, X_2, \ldots, X_n$ uniformes em $[0, 2\pi]$ e função $f(X_1, \ldots, X_n) = \sum_{i=1}^{n} sen(X_i) = X$. Nesse caso, $c = 2$.

$$\mathbb{P}\left(\left|X - \mathbb{E}(X)\right| > t \cdot n\right) \leq 2 \cdot \exp\left\{-(t^2/2) \cdot n\right\} \quad \text{(McDiarmid} \equiv \text{Chernoff)}$$

Desigualdade de McDiarmid (Bounded Differences)

Método das Diferenças Limitadas: *generaliza Chernoff-Hoeffding.*
v.a. **independentes** $X_1, \ldots, X_n$ e função $f(x_1, \ldots, x_n)$ c-Lipschitz.

$$\mathbb{P}\Big(\Big|f(X_1, \ldots, X_n) - \mathbb{E}(f(X_1, \ldots, X_n))\Big| \; > \; t \cdot n\Big) \; \leq \; 2 \cdot \exp\left\{\frac{-2t^2}{c^2} \cdot n\right\}$$

Exemplo: Grafo aleatório $G \sim G(n, p)$ de Erdős-Rényi:
n vértices $\{1, 2, \ldots, n\}$ e uma v.a. $X_{i,j} \sim \mathit{Bernoulli}(p)$ para cada par (i, j) de vértices distintos.

Considerar outra sequência de v.a.: $Y_2, Y_3, \ldots, Y_n$, onde Y_i representa as v.a.'s $X_{1,i}, \ldots, X_{i-1,i}$. Seja Y o tamanho da maior clique $\omega(G)$ (ou o número mínimo de cores $\chi(G)$).

$$\mathbb{P}\Big(\Big|Y - \mathbb{E}(Y)\Big| \; > \; t \cdot n\Big) \; \leq \; 2 \cdot \exp\left\{-2t^2 \cdot n\right\} \qquad \text{(McDiarmid)}$$

Desigualdade de McDiarmid (generalização)

Função $(c_1, \ldots, c_n)$-Lipschitz: Função $f(x_1, \ldots, x_n)$ tal que $|f(x) - f(x')| \leq c_i$ se x e x' diferem apenas na coordenada i.

Método das Diferenças Limitadas: *generaliza Chernoff-Hoeffding.*
v.a. **independentes** $X_1, \ldots, X_n$ e
função $f(x_1, \ldots, x_n)$ $(c_1, \ldots, c_n)$-Lipschitz.

$$\mathbb{P}\left(\left|f(X_1, \ldots, X_n) - \mathbb{E}(f(X_1, \ldots, X_n))\right| > t \cdot n\right) \leq 2 \cdot \exp\left\{\frac{-2t^2}{\sum_{i=1}^{n} c_i^2} \cdot n^2\right\}$$

Chernoff-Hoeffding (generalização): só serve p/ somatório de v.a.'s
v.a.'s **independentes** $X_i \in [a_i, b_i]$. Seja $X = \sum_{i=1}^{n} X_i$.

$$\mathbb{P}\left(|X - \mathbb{E}(X)| > t \cdot n\right) \leq 2 \cdot \exp\left\{\frac{-2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2} \cdot n^2\right\}$$

Convergência em probabilidade (w.h.p)

Seja $A(n)$ um evento em um espaço de probabilidade, onde n é um parâmetro. Dizemos que $A(n)$ ocorre "*com alta probabilidade*" (w.h.p) se:

$$\lim_{n\to\infty} \mathbb{P}(A_n) \;=\; 1 \qquad \text{(convergência em probabilidade)}.$$

Por exemplo,

$$\mathbb{P}(A(n)) \;\geq\; 1 - \frac{1}{n^{\beta}} \qquad \text{para algum } \beta > 0 \text{ constante}$$

Exemplo: v.a. X, que é a soma de n variáveis 0/1 X_i com prob. $1/2$ de ser 1. $\mathbb{E}(X) = n/2$. Onde está concentrada a distribuição de X? Determinar quando $\mathbb{P}(|X - \mathbb{E}(X)| \geq \alpha) < 1/n$ (w.h.p: probabilidade da "cauda" tende a 0).

$$\mathbb{P}(|X - \mathbb{E}(X)| \geq \alpha) \leq 2 \cdot \exp\left\{\frac{-\alpha^2}{3\mathbb{E}(X)}\right\} = 2 \cdot \exp\left\{\frac{-2\alpha^2}{3n}\right\} < 1/n$$

Tomando $\alpha \geq 2\sqrt{n \cdot \ln n}$ e n suficientemente grande, temos o desejado.

Leis dos Grandes Números

Seja $X_1, X_2, X_3, \ldots$ uma sequência de v.a. iid (independentes e identicamente distribuídas) com média μ e variância σ^2 finitos.
A **Média Amostral** M_n é definida como $M_n = \frac{1}{n} \cdot \sum_{i=1}^{n} X_i$

$$\mathbb{E}(M_n) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i) = \frac{1}{n}\cdot n\mu = \mu$$

$$\mathbb{V}ar(M_n) = \mathbb{V}ar\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\sum_{i=1}^{n}\mathbb{V}ar(X_i) = \frac{1}{n^2}\cdot n\sigma^2 = \frac{\sigma^2}{n}$$

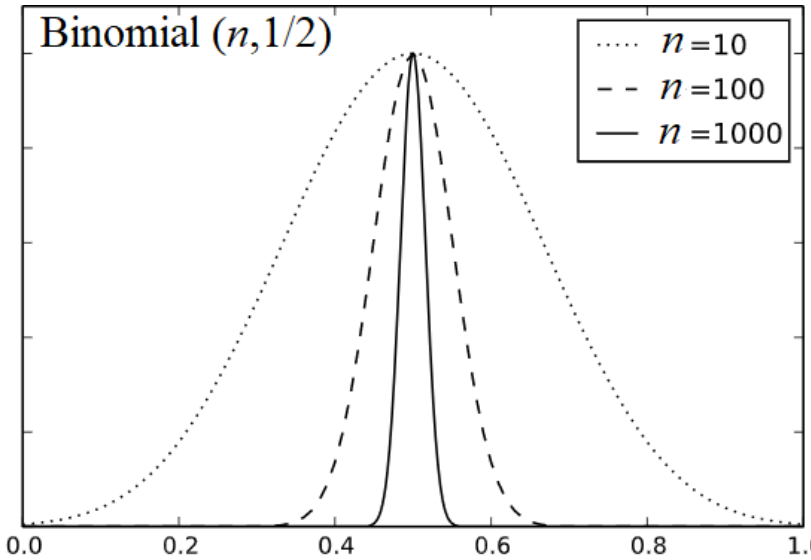
Conclusão: A Média Amostral M_n tem mesma média μ e variância que tende a 0 quanto maior for o número n de "amostras".

Teorema Central do Limite: M_n converge em distribuição para uma v.a. de distribuição Normal com média μ e variância σ^2/n.

Lei Fraca dos Grandes Números: Para todo $\varepsilon > 0$, $|M_n - \mu| < \varepsilon$ w.h.p. Assim, $\lim_{n\to\infty} \mathbb{P}(|M_n - \mu| < \varepsilon) = 1$ (convergência em probabilidade).

Lei Forte dos Grandes Números: $\mathbb{P}(\lim_{n\to\infty} M_n = \mu) = 1$ (convergência quase certa). M_n realmente converge p/ valor esperado μ.

Leis dos Grandes Números



M_n fica mais concentrada

Conclusão: A Média Amostral M_n tem mesma média μ e variância que tende a 0 quanto maior for o número n de "amostras".

Teorema Central do Limite: M_n converge em distribuição para uma v.a. de distribuição Normal com média μ e variância σ^2/n.

Lei Fraca dos Grandes Números: Para todo $\varepsilon > 0$, $|M_n - \mu| < \varepsilon$ w.h.p. Assim, $\lim_{n\to\infty} \mathbb{P}(|M_n - \mu| < \varepsilon) = 1$ (convergência em probabilidade).

Lei Forte dos Grandes Números: $\mathbb{P}(\lim_{n\to\infty} M_n = \mu) = 1$ (convergência quase certa). M_n realmente converge p/ valor esperado μ.

Margem de Erro e Confiança

Dizemos que um experimento tem Margem de Erro ε e Confiança β se a Média Amostral M_n (com valor esperado μ) satisfaz:

$$\mathbb{P}\left(M_n = \mu \pm \varepsilon\right) \;=\; \mathbb{P}\left(|M_n - \mu| \leq \varepsilon\right) \;\geq\; \beta$$

Lembrando que $\mathbb{V}ar(M_n) = \sigma^2/n$, temos por Chebyshev que

$$\mathbb{P}(M_n = \mu \pm \varepsilon) \;\geq\; 1 - \frac{\sigma^2}{\varepsilon^2 n} \;=\; \beta,$$

tomando

$$n \;=\; \frac{\sigma^2}{\varepsilon^2(1-\beta)}.$$

Diminuir a Margem de Erro (ou aumentar a precisão) tem mais impacto do que aumentar a Confiança.

Exemplo: Testar com margem de erro 1% e confiança 99% se uma moeda é viciada. Seja M_n a média amostral de caras.

$$\mathbb{P}(M_n \;=\; 0.5 \pm 0.01) \;\geq\; 1 - \frac{0.25}{0.01^2 n} \;=\; 0.99;$$

tomando $n = 250.000$

Margem de Erro e Confiança

Exemplo: Pesquisa com 2000 eleitores indica que certo candidato vence no 2o turno com 55% de votos. Calcule a margem de erro para 95% de confiança.

Considerando cada pessoa como uma v.a. $X_i \sim Bernoulli(0.55)$, temos: $\mu = 0.55$ e $\sigma^2 = 0.55 \cdot 0.45$

$$\mathbb{P}(M_n = \mu \pm \varepsilon) \geq 1 - \frac{\sigma^2}{\varepsilon^2 n} = \beta,$$

$$\mathbb{P}(M_n = 0.55 \pm \varepsilon) \geq 1 - \frac{0.55 \cdot 0.45}{\varepsilon^2 \cdot 2000} = 0.95;$$

tomando $\varepsilon = 0,0498 = 4,98\%$.

Margem de Erro e Confiança

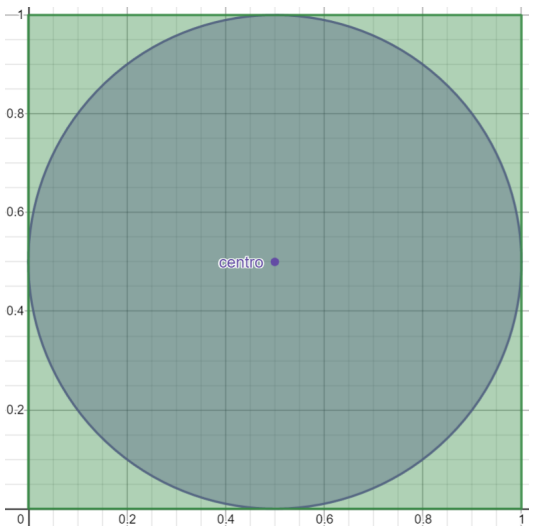
Exemplo: Calcular o valor de π experimentalmente.

Sortear n pontos unif aleatórios no quadrado $[0,1]^2$ e ver quantos caem no círculo de diâmetro 1 (com área $\pi/4$) dentro do quadrado.
Acertar a 4o casa decimal (margem de erro 10^{-5}) com confiança 70%.
$\mu = \pi/4$ e $\sigma^2 = \frac{\pi}{4}(1 - \frac{\pi}{4})$.

$$\mathbb{P}(M_n = \mu \pm \varepsilon) \;\geq\; 1 - \frac{\sigma^2}{\varepsilon^2 n} \;=\; \beta,$$

$$\mathbb{P}\left(4 \cdot M_n = \pi \pm 4 \cdot 10^{-5}\right) = \mathbb{P}\left(M_n = \frac{\pi}{4} \pm 10^{-5}\right) \;\geq\; 1 - \frac{\pi \cdot (4-\pi)}{4^2 \cdot 10^{-10} \cdot n} \;=\; 0.7;$$

tomando $n = 6 \cdot 10^9$ (seis bilhões de experimentos).



MCM (Monte Carlo Method)

- Algoritmos que realizam muitas amostragens aleatórias para obter soluções razoáveis de problemas difíceis.
- Pelas Leis dos Grandes Números, uma grande quantidade de amostras levam à solução do problema (no limite tendendo ao infinito).

Algoritmo para Elemento Máximo de um vetor

Máximo (vetor A, inteiro n)

```
1  max ← 0
2  para i ← 1 até n faça:
3        se A[i] > max então
4              max ← A[i]
5  retorne max
```

Pergunta: Quantas vezes a linha 4 é executada no caso médio?

Suponha que o vetor A é uma permutação aleatória uniforme de 1 a n.
Cada permutação tem probabilidade $1/n!$

Seja $X_i = 1$ se a linha 4 foi executada na iteração i; e 0, cc.
Seja X o número total de execuções da linha 4: $X = \sum_{i=1}^{n} X_i$.

Qual a esperança $\mathbb{E}(X_i)$? É a prob da linha 4 ser executada na iteração i.

Resposta: $\mathbb{E}(X_i) = 1/i$, pois o elemento $A[i]$ deveria ser o maior entre os primeiros i elementos $A[1 \ldots i]$.

Algoritmo para Elemento Máximo de um vetor

Máximo (vetor A, inteiro n)

```
1  max ← 0
2  para i ← 1 até n faça:
3        se A[i] > max então
4              max ← A[i]
5  retorne max
```

Pergunta: Quantas vezes a linha 4 é executada no caso médio?

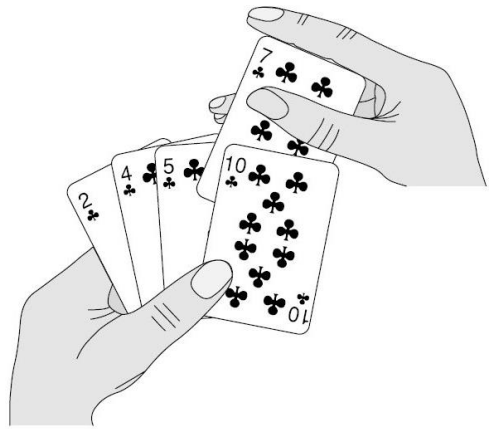
Seja $X_i = 1$ se a linha 4 foi executada na iteração i; $\mathbb{E}(X_i) = 1/i$
Seja X o número total de execuções da linha 4: $X = \sum_{i=1}^{n} X_i$.

$$\mathbb{E}(X) = \mathbb{E}\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} \mathbb{E}(X_i) = \sum_{i=1}^{n} \frac{1}{i} \approx \ln n + 0.5772 + \frac{1}{2n}$$

Análise do Caso Médio do Insertion Sort

Insertion-Sort (vetor A, inteiro n)

```
1  para k ← 2 até n faça:
2        carta ← A[j];          i ← k − 1
3        enquanto i ≥ 1 e A[i] > carta faça:
4              A[i + 1] ← A[i];     i ← i − 1
5        A[i + 1] ← carta
```



Tempo do Caso Médio: Considere o vetor de entrada do algoritmo como uma permutação de 1 a n escolhida aleatoriamente com prob uniforme.
É na linha 3 onde são feitas comparações entre elementos do vetor.
Qual é o número esperado de execuções da linha 3?

Para cada valor de k, a linha 3 é executada um num de vezes entre 1 e k.
Para um certo t entre 1 e k, qual a probab de ser executada t vezes?

Resposta: $1/k$, pois a nova carta (a k-ésima) deveria ser a t-ésima maior entre as primeiras k cartas $A[1 \ldots k]$.

Análise do Caso Médio do Insertion Sort

Tempo do Caso Médio: Considere o vetor de entrada do algoritmo como uma permutação de 1 a n escolhida aleatoriamente com probabilidade uniforme. Qual é o número esperado de execuções da linha 3?

Para cada valor de k, a linha 3 é executada um num de vezes entre 1 e k. Para um certo t entre 1 e k, qual a probab de ser executada t vezes?

Resposta: $1/k$, pois a nova carta (a k-ésima) deveria ser a t-ésima maior entre as primeiras k cartas $A[1 \ldots k]$.

Seja X_k o número de execuções da linha 3 para um certo k fixo. Portanto:

$$\mathbb{E}(X_k) = \sum_{t=1}^{k} t \cdot \mathbb{P}(X_k = t) = \sum_{t=1}^{k} t \cdot \frac{1}{k} = \frac{1}{k} \sum_{t=1}^{k} t = \frac{1}{k} \cdot \frac{k(k+1)}{2} = \frac{k+1}{2}$$

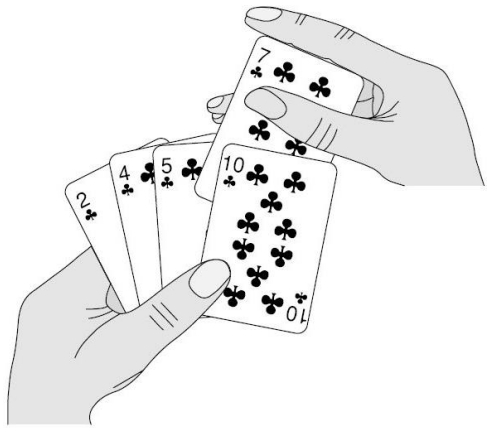
Seja X o número total de execuções da linha 3: $X = \sum_{k=2}^{n} X_k$. Portanto:

$$\mathbb{E}(X) = \mathbb{E}\left(\sum_{k=2}^{n} X_k\right) = \sum_{k=2}^{n} \mathbb{E}(X_k) = \sum_{k=2}^{n} \frac{k+1}{2} = \frac{(n+4)(n-1)}{4} = \Theta(n^2)$$

Insertion Sort Probabilístico

Insertion-Sort-Prob (vetor A, inteiro n)

1. **rearranja** o vetor A de modo aleatório e uniforme
2. Insertion-Sort(A,n)



Algoritmo Las Vegas de ordenação (sempre produz a resposta esperada)

Tempo Esperado do Insertion Sort Probabilístico supondo todos os elementos diferentes entre si no vetor A: análise praticamente idêntica à do Tempo do Caso Médio do Insertion Sort Determinístico.

Tempo Esperado $\Theta(n^2)$

Análise do Caso Médio do Quick Sort

Quick-Sort (vetor A, inteiros p, r)

```
1  se p < r então:
2      q ← Particione(A, p, r)
3      Quick-Sort(A,p,q − 1)
4      Quick-Sort(A,q + 1,r)
```

Particione (vetor A, inteiros p, r)

```
1  pivo ← A[r];    i ← p − 1
2  para k ← p até r faça:
3      se A[k] ≤ pivo então
4          i ← i + 1; Trocar A[i] ↔ A[j]
5  retorne i
```

Tempo do Caso Médio: Considere o vetor de entrada do algoritmo como uma permutação de 1 a n escolhida aleatoriamente com prob uniforme.
É na linha 3 do Particione onde são feitas comparações entre elementos.
Qual é o número esperado de execuções da linha 3 do Particione?
Boa estimativa para o tempo total do Quick Sort.

Análise do Caso Médio do Quick Sort

Quick-Sort (vetor A, inteiros p, r)

1 **se** $p < r$ **então**:
2 $\quad q \leftarrow Particione(A, p, r)$
3 $\quad$ Quick-Sort(A,p,$q-1$)
4 $\quad$ Quick-Sort(A,$q+1$,r)

Particione (vetor A, inteiros p, r)

1 $pivo \leftarrow A[r]; \quad i \leftarrow p-1$
2 **para** $k \leftarrow p$ **até** r **faça**:
3 $\quad$ **se** $A[k] \leq pivo$ **então**
4 $\quad\quad i \leftarrow i+1$; Trocar $A[i] \leftrightarrow A[j]$
5 **retorne** i

Tempo do Caso Médio: Na linha 3 do Particione, os elementos são comparados com o pivô, que depois nunca mais será analisado.
Para $a, b \in \{1 \ldots, n\}$, seja $X_{a,b} = 1$ se os elementos com valor a e b no vetor são comparados na linha 3 do Particione.
Seja X o número total de execuções da linha 3: $X = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} X_{a,b}$.

$$\mathbb{E}(X_{a,b}) \;=\; \mathbb{P}(a \text{ ou } b \text{ ser o 1o pivô em } [a,b]) \;=\; \frac{2}{b-a+1}$$

$$\mathbb{E}(X) = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \mathbb{E}(X_{a,b}) = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \frac{2}{b-a+1} \leq \sum_{a=1}^{n-1} \sum_{\ell=1}^{n} \frac{2}{\ell} \leq 2n(\ln n + 1)$$

Quick Sort Probabilístico

Quick-Sort-Prob (A, p, r)

```
1  se p < r então:
2      q ← Particione-Aleat(A, p, r)
3      Quick-Sort-Prob(A,p,q − 1)
4      Quick-Sort-Prob(A,q + 1,r)
```

Particione-Aleat (A, p, r)

```
1  i ← random(p, r)
2  Trocar A[i] ↔ A[r]
3  Particione(A,p,r)
```

Algoritmo Las Vegas de ordenação (sempre produz a resposta esperada)

Tempo Esperado do Quick Sort Probabilístico supondo todos os elementos diferentes entre si no vetor A: análise praticamente idêntica à do Tempo do Caso Médio do Quick Sort Determinístico.

Tempo Esperado $\Theta(n \cdot \log n)$

Análise do Caso Médio da Seleção do k-ésimo mínimo

Seleção (vetor A, inteiros p, r, k)

1 **se** $p = r$ **retorne** $A[p]$
2 $q \leftarrow Particione(A, p, r)$
3 **se** $k = q - p + 1$ **então**
4 **retorne** $A[q]$
5 **senão se** $k < q - p + 1$ **então**
6 **retorne** Seleção $(A, p, q-1, k)$
7 **senão retorne** Seleção $(A, q+1, r, k-(q-p+1))$

Particione (vetor A, inteiros p, r)

1 $pivo \leftarrow A[r]$; $i \leftarrow p - 1$
2 **para** $k \leftarrow p$ **até** r **faça**:
3 **se** $A[k] \leq pivo$ **então**
4 $i \leftarrow i + 1$; Trocar $A[i] \leftrightarrow A[j]$
5 **retorne** i

Tempo do Caso Médio: Considere o vetor de entrada do algoritmo como uma permutação de 1 a n escolhida aleatoriamente com prob uniforme.
É na linha 3 do Particione onde são feitas comparações entre elementos.
Qual é o número esperado de execuções da linha 3 do Particione?
Boa estimativa para o tempo total da Seleção do k-ésimo mínimo.

Análise do Caso Médio da Seleção do k-ésimo mínimo

```
Seleção (vetor A, inteiros p, r, k)
1  se p = r retorne A[p]
2  q ← Particione(A, p, r)
3  se k = q − p + 1 então
4      retorne A[q]
5  senão se k < q − p + 1 então
6      retorne Seleção (A,p,q − 1,k)
7  senão retorne
   Seleção (A,q + 1,r,k − (q − p + 1))
```

```
Particione (vetor A, inteiros p, r)
1  pivo ← A[r];    i ← p − 1
2  para k ← p até r faça:
3      se A[k] ≤ pivo então
4          i ← i + 1; Trocar A[i] ↔ A[j]
5  retorne i
```

Tempo do Caso Médio: Na linha 3 do Particione, os elementos são comparados com o pivô, que depois nunca mais será analisado. Para $a, b \in \{1 \ldots, n\}$, seja $X_{a,b} = 1$ se os elementos com valor a e b no vetor são comparados na linha 3 do Particione.

$$\mathbb{E}(X_{a,b}) \;=\; \mathbb{P}(a \text{ ou } b \text{ ser o 1o pivô em } [a, b, k]).$$

Por exemplo, se $k < a$ e o 1o pivô em $[k, b]$ não for a ou b, nunca serão comparados pois ficarão em lados opostos entre si ou em lados opostos com relação ao k.

Análise do Caso Médio da Seleção do k-ésimo mínimo

Tempo do Caso Médio:

$\mathbb{E}(X_{a,b}) = \mathbb{P}(a \text{ ou } b \text{ ser o 1o pivô em } [a, b, k])$.

Por exemplo, se $k < a$ e o 1o pivô em $[k, b]$ não for a ou b, nunca serão comparados pois ficarão em lados opostos entre si ou em lados opostos com relação ao k.

$\mathbb{E}(X_{a,b}) = \frac{2}{b-k+1}$, se $a \geq k$
$\mathbb{E}(X_{a,b}) = \frac{2}{k-a+1}$, se $b \leq k$
$\mathbb{E}(X_{a,b}) = \frac{2}{b-a+1}$, se $a < k < b$

Seja X o número total de execuções da linha 3: $X = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} X_{a,b}$.
Para facilitar as contas, vamos assumir que $k = 1$

$$\mathbb{E}(X) = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} \mathbb{E}(X_{a,b}) = \sum_{b=2}^{n} \sum_{a=1}^{b-1} \frac{2}{b} = \sum_{b=2}^{n} \frac{2(b-1)}{b} \leq 2n$$

Análise do Caso Médio da Seleção do k-ésimo mínimo

Tempo do Caso Médio: $\mathbb{E}(X_{a,b}) = \mathbb{P}(a \text{ ou } b \text{ ser o 1o pivô em } [a, b, k])$.

Por exemplo, se $k < a$ e o 1o pivô em $[k, b]$ não for a ou b, nunca serão comparados pois ficarão em lados opostos entre si ou em lados opostos com relação ao k.

$\mathbb{E}(X_{a,b}) = \frac{2}{b-k+1}$, se $a \geq k$
$\mathbb{E}(X_{a,b}) = \frac{2}{k-a+1}$, se $b \leq k$
$\mathbb{E}(X_{a,b}) = \frac{2}{b-a+1}$, se $a < k < b$

Seja X o número total de execuções da linha 3: $X = \sum_{a=1}^{n-1} \sum_{b=a+1}^{n} X_{a,b}$.

$$\sum_{a=k}^{n-1} \sum_{b=a+1}^{n} \mathbb{E}(X_{a,b}) = \sum_{b=k+1}^{n} \sum_{a=k}^{b-1} \frac{2}{b-k+1} = \sum_{b=k+1}^{n} \frac{2(b-k)}{b-k+1} \leq 2(n-k)$$

$$\sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \mathbb{E}(X_{a,b}) = \sum_{a=1}^{k-1} \sum_{b=a+1}^{k} \frac{2}{k-a+1} = \sum_{a=1}^{k-1} \frac{2(k-a)}{k-a+1} \leq 2(k-1)$$

$$\sum_{a=1}^{k-1} \sum_{b=k+1}^{n} \mathbb{E}(X_{a,b}) = \sum_{a=1}^{k-1} \sum_{b=k+1}^{n} \frac{2}{b-a+1} \leq \sum_{\ell=3}^{n} \sum_{a=k-\ell+2}^{k-1} \frac{2}{\ell} \leq \sum_{\ell=3}^{n} \ell \cdot \frac{2}{\ell} \leq 2n$$

Seleção Probabilística do k-ésimo mínimo

Seleção-Prob (vetor A, inteiros p, r, k)

```
1  se p = r retorne A[p]
2  q ← Particione-Aleat(A, p, r)
3  se k = q − p + 1 então
4      retorne A[q]
5  senão se k < q − p + 1 então
6      retorne Seleção-Prob (A,p,q − 1,k)
7  senão retorne
           Seleção-Prob (A,q + 1,r,k − (q − p + 1))
```

Particione-Aleat (A, p, r)

```
1  i ← random(p, r)
2  Trocar A[i] ↔ A[r]
3  Particione(A,p,r)
```

Algoritmo Las Vegas de seleção (sempre produz a resposta esperada)

Tempo Esperado da Seleção Probabilística supondo todos os elementos diferentes entre si no vetor A: análise praticamente idêntica à do Tempo do Caso Médio da Seleção Determinística.

Tempo Esperado $\Theta(n)$

Cadeias de Markov (*Markov Chains*)

Um **processo estocástico discreto** é uma sequência $X_0, X_1, X_2, \ldots$ de v.a. que assumem valores em um conjunto finito (ou infinito enumerável). Se $X_t = i$, dizemos que o processo está no estado i no tempo t.

Uma **cadeia de Markov** é um processo estocástico discreto tal que $\mathbb{P}(X_t = a_t \mid X_{t-1} = a_{t-1}, \ldots, X_0 = a_0) = \mathbb{P}(X_t = a_t \mid X_{t-1} = a_{t-1})$, cujo valor não depende de t (*Memoryless/Markov property*).

Seja $P_{i,j} = \mathbb{P}(X_{t+1} = j \mid X_t = i)$.

Matriz de transição (1 passo): $$P = \begin{bmatrix} P_{0,0} & P_{0,1} & \ldots & P_{0,j} & \ldots \\ P_{1,0} & P_{1,1} & \ldots & P_{1,j} & \ldots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ P_{i,0} & P_{i,1} & \ldots & P_{i,j} & \ldots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

Lei da Probabilidade total: $\sum_{j=0}^{\infty} P_{i,j} = 1$

Markov Chain - transição em m passos

Matriz de transição (m passos): Seja $P_{i,j}^{(m)} = \mathbb{P}(X_{t+m} = j \mid X_t = i)$.

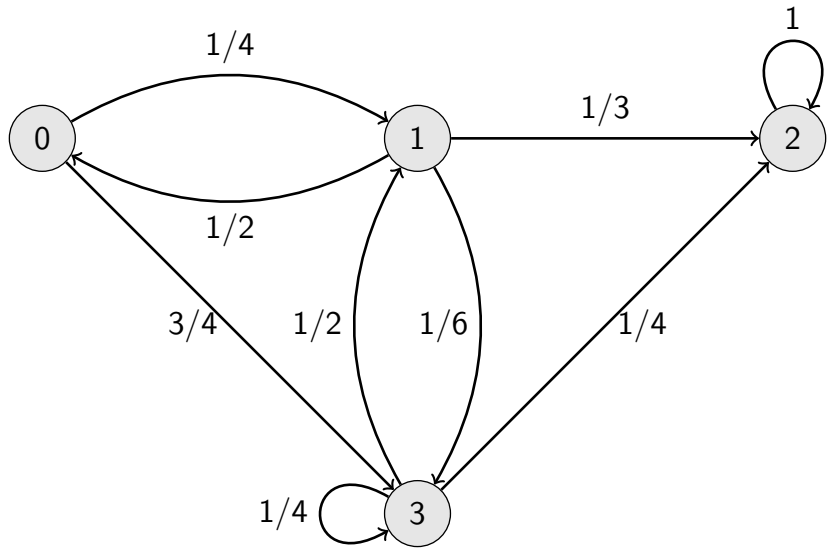
$$P_{i,j}^{(m)} = \sum_{k=0}^{\infty} \mathbb{P}(X_{t+m} = j, X_{t+1} = k \mid X_t = i) = \sum_{k=0}^{\infty} \mathbb{P}(X_{t+m} = j \mid X_{t+1} = k, X_t =$$

$$P_{i,j}^{(m)} = \sum_{k=0}^{\infty} P_{i,k} \cdot P_{k,j}^{(m-1)}$$

Seja $P^{(m)}$ a matriz com valores [$P_{i,j}^{(m)}$]. Logo $P^{(m)} = P \cdot P^{(m-1)}$.
Por indução, temos que $P^{(m)} = P^m$.

Matriz transição (m passos): $P^{(m)} = \begin{bmatrix} P_{0,0}^{(m)} & P_{0,1}^{(m)} & \cdots & P_{0,j}^{(m)} & \cdots \\ P_{1,0}^{(m)} & P_{1,1}^{(m)} & \cdots & P_{1,j}^{(m)} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ P_{i,0}^{(m)} & P_{i,1}^{(m)} & \cdots & P_{i,j}^{(m)} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix} = P^m$

Markov Chain - transição em m passos - exemplo

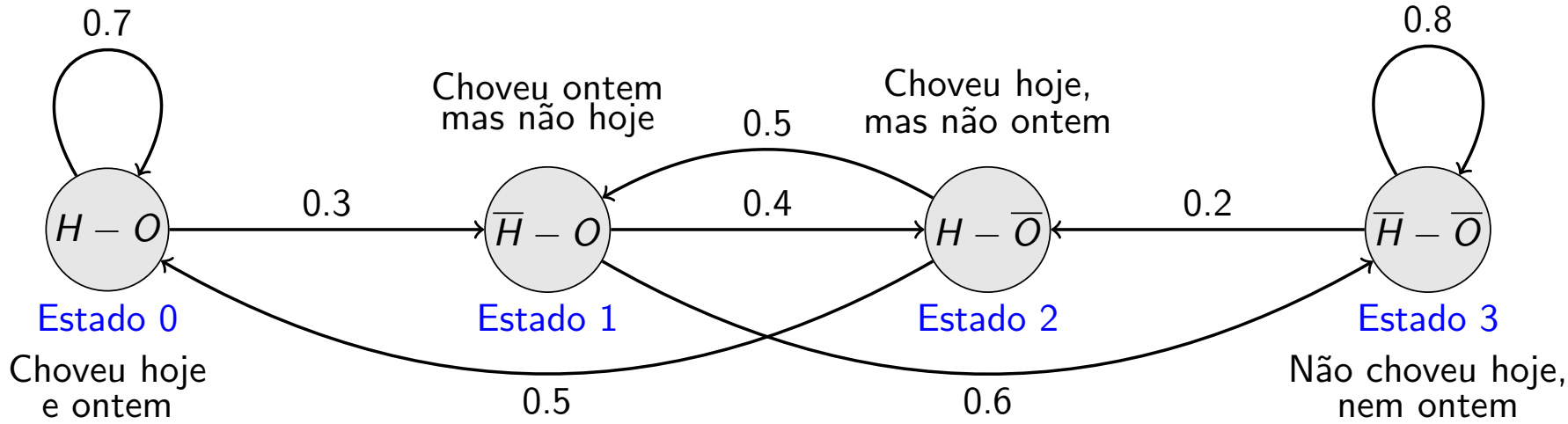


$$P = \begin{bmatrix} 0 & 1/4 & 0 & 3/4 \\ 1/2 & 0 & 1/3 & 1/6 \\ 0 & 0 & 1 & 0 \\ 0 & 1/2 & 1/4 & 1/4 \end{bmatrix} \quad \Rightarrow \quad P^3 = \begin{bmatrix} 3/16 & 7/48 & 29/64 & 41/192 \\ 5/48 & 5/24 & 79/144 & 5/36 \\ 0 & 0 & 1 & 0 \\ 1/16 & 13/96 & 107/192 & 47/192 \end{bmatrix}$$

$P_{0,3}^{(3)} = 41/192$: prob de ir de 0 a 3 em 3 passos.
Outro modo: caminhos possíveis 0103, 0133, 0313, 0333.

$$\frac{1}{4}\cdot\frac{1}{2}\cdot\frac{3}{4} + \frac{1}{4}\cdot\frac{1}{6}\cdot\frac{1}{4} + \frac{3}{4}\cdot\frac{1}{2}\cdot\frac{1}{6} + \frac{3}{4}\cdot\frac{1}{4}\cdot\frac{1}{4} = \frac{3}{32} + \frac{1}{96} + \frac{1}{16} + \frac{3}{64} = \frac{41}{192}$$

Markov Chain - transição em m passos - exemplo



$$P = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.2 & 0.8 \end{bmatrix} \quad \Rightarrow \quad P^2 = \begin{bmatrix} 0.49 & 0.21 & 0.12 & 0.18 \\ 0.20 & 0.20 & 0.12 & 0.48 \\ 0.35 & 0.15 & 0.20 & 0.30 \\ 0.10 & 0.10 & 0.16 & 0.64 \end{bmatrix}$$

Choveu na segunda e na terça. Qual a probabilidade de chover na quinta?

$P_{0,0}^{(2)} + P_{0,2}^{(2)} = 0.49 + 0.12 = 0.61$

Markov Chain - Classificação dos estados

Dizemos que o Estado j é **acessível** a partir do Estado i se:

$$P_{i,j}^{(n)} > 0 \text{ para algum } n \geq 0.$$

Dizemos que os Estados i e j **se comunicam** ($i \leftrightarrow j$) se

i é acessível de j e j é acessível de i.

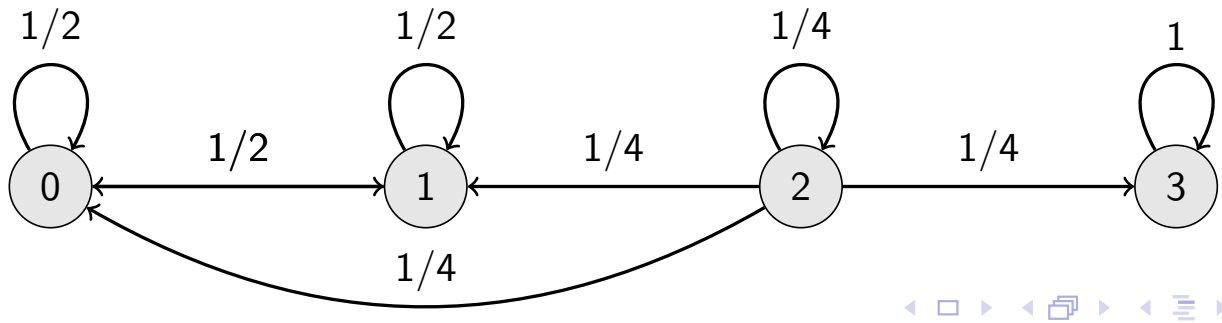
Propriedades: (Relação de Equivalência)

- **Reflexiva:** $i \leftrightarrow i$.
- **Simétrica:** se $i \leftrightarrow j$, então $j \leftrightarrow i$.
- **Transitiva:** se $i \leftrightarrow j$ e $j \leftrightarrow k$, então $i \leftrightarrow k$.

Partição dos Estados da Cadeia de Markov em Classes de Equivalência com relação à comunicabilidade. Mostrar 2 exemplos slides anteriores.

Uma cadeia é **irredutível** se possui apenas 1 classe.

Outro exemplo: 3 classes: $\{0,1\}, \quad \{2\}, \quad \{3\}$



Markov Chain - Estados Recorrentes $\times$ Estados Transientes

Seja f_{ij} a prob de, começando no Estado i, o processo visitar j em algum momento futuro. Ou seja, $1 - f_{ii}$ é a prob de nunca mais voltar.

Dizemos que o Estado i é **recorrente** se $f_{ii} = 1$. Caso contrário, $f_{ii} < 1$ e o estado é **transiente** (existe a prob $1 - f_{ii}$ de nunca mais voltar).

Se um estado i é transiente, o número X de vezes que o processo esteve em i, começando em i, é uma v.a. com distribuição geométrica: $\mathbb{P}(X = n) = f_{ii}^n \cdot (1 - f_{ii})$.

Proposição 4.3.1 (Ross):

$$\text{Um Estado } i \text{ é transiente} \quad \textbf{se e só se} \quad \sum_{n=0}^{\infty} P_{i,i}^{(n)} < \infty.$$

Corolário 4.3.1 (Ross):
Se o Estado i é recorrente e $i \leftrightarrow j$, então o Estado j é recorrente.

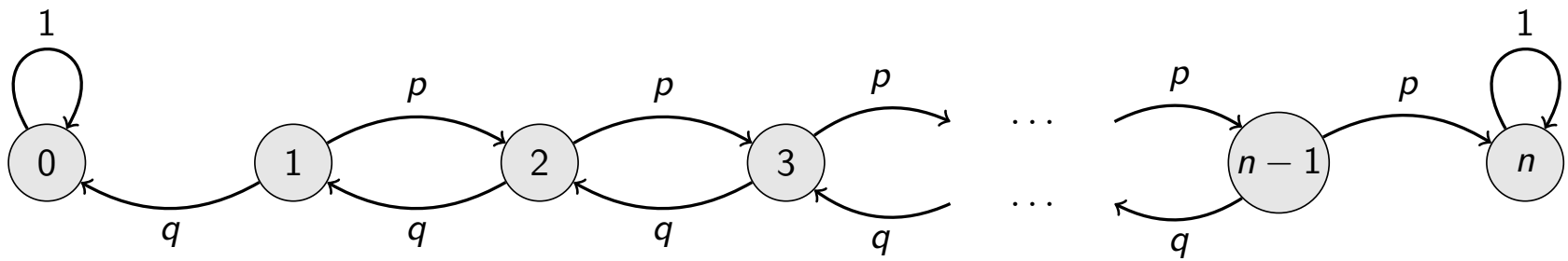
Observação: Toda cadeia de Markov finita tem um estado recorrente.

Markov Chain - Ruína do Jogador (*Gambler's Ruin*)

Probabilidade p de ganhar 1 real e prob $q = 1 - p$ de perder 1 real. Jogadas independentes. Qual a probabilidade de conseguir n reais começando com 1 real?

Modelar o problema como uma Cadeia de Markov:

$P_{0,0} = P_{n,n} = 1$; $P_{i,i+1} = p$; $P_{i,i-1} = q$ para $i = 1, \ldots, n-1$.



3 classes: $\{0\}$ recorrente, $\{n\}$ recorrente, $\{1, \ldots, n-1\}$ transiente.

Seja R_i a prob de, começando com i reais, atingir n reais antes de 0:

$R_0 = 0$; $R_n = 1$; $R_i = p \cdot R_{i+1} + q \cdot R_{i-1}$ para $i = 1, \ldots, n-1$.

Objetivo: Calcular R_1. Generalizar para R_i.

Markov Chain - Ruína do Jogador (*Gambler's Ruin*)

Seja R_i a prob de, começando com i reais, atingir n reais antes de 0:
$R_0 = 0; \quad R_n = 1; \quad R_i = p \cdot R_{i+1} + q \cdot R_{i-1}$ para $i = 1, \ldots, n-1$.

Objetivo: Calcular R_1. Generalizar para R_i.

$$R_{i+1} - R_i = \left(\frac{q}{p}\right) \cdot (R_i - R_{i-1}) = \left(\frac{q}{p}\right)^2 \cdot (R_{i-1} - R_{i-2}) = \left(\frac{q}{p}\right)^i \cdot (R_1 - R_0) = \left(\frac{q}{p}\right)^i \cdot R_1$$

$$\Rightarrow R_{i+1} = R_i + \left(\frac{q}{p}\right)^i R_1 \Rightarrow R_i = R_1 \cdot \left[1 + \frac{q}{p} + \left(\frac{q}{p}\right)^2 + \ldots + \left(\frac{q}{p}\right)^{i-1}\right]$$

Se $p = 1/2$: $R_i = i \cdot R_1$. Se $p \neq 1/2$: $R_i = \frac{1-(q/p)^i}{1-(q/p)} \cdot R_1$.

Como $R_n = 1$, então:

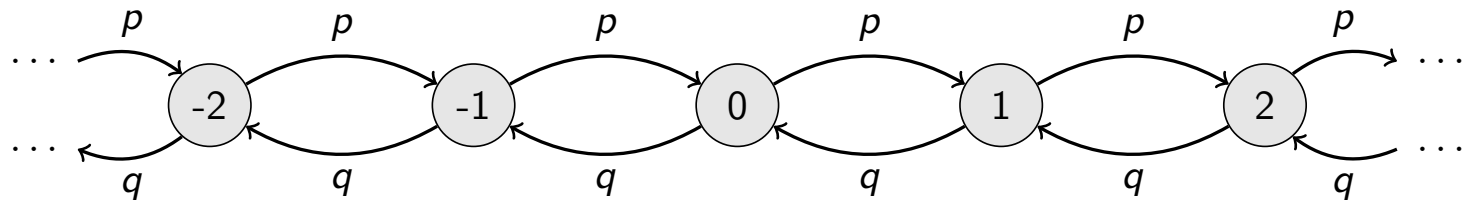
Se $p = 1/2$: $R_n = n \cdot R_1$. Se $p \neq 1/2$: $R_n = \frac{1-(q/p)^n}{1-(q/p)} \cdot R_1$. **Portanto:**

Se $p = 1/2$: $R_1 = 1/n$. Se $p \neq 1/2$: $R_1 = \frac{1-(q/p)}{1-(q/p)^n}$. **Finalmente:**

Se $p = 1/2$: $R_i = i/n$. Se $p \neq 1/2$: $R_i = \frac{1-(q/p)^i}{1-(q/p)^n}$.

Markov Chain - Passeio Aleatório (*Random Walk*)

$P_{i,i+1} = p$; $P_{i,i-1} = q = 1 - p$; onde $i \in \mathbb{Z}$.



Todos os estados **se comunicam**. Logo, todos são recorrentes ou todos são transientes. O estado 0 é recorrente **se e só se** $\sum_{n=0}^{\infty} P_{0,0}^{(n)} = \infty$.

$$P_{0,0}^{(2n+1)} = 0, \ \forall n. \qquad P_{0,0}^{(2n)} = \binom{2n}{n} p^n (1-p)^n = \frac{(2n)!}{n!n!}[p(1-p)]^n \approx \frac{[4p(1-p)]^n}{\sqrt{\pi \cdot n}},$$

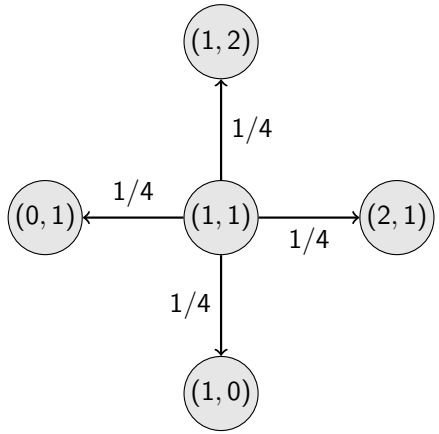
usando Stirling: $n! \approx n^n e^{-n}\sqrt{2\pi n}$. Note que $4p(1-p) \leq 1$, $\forall p \in [0,1]$.

Igualdade $4p(1-p) = 1$ se e só se $p = 1/2$.

Logo, $\sum_{n=0}^{\infty} P_{0,0}^{(2n)} = \sum_{n=0}^{\infty} \frac{[4p(1-p)]^n}{\sqrt{\pi \cdot n}} = \infty$ **se e só se** $p = 1/2$.

Conclusão: Se $p = 1/2$, todos os estados são recorrentes.
Se $p \neq 1/2$, todos os estados são transientes.

Markov Chain - Random Walk em 2 dimensões



$$P_{1,1}^{(2n)} = \sum_{i=0}^{n} \frac{(2n)! \cdot (1/4)^{2n}}{i! \cdot i! \cdot (n-i)! \cdot (n-i)!} = \left(\frac{1}{4}\right)^{2n} \binom{2n}{n} \sum_{i=0}^{n} \binom{n}{i} \binom{n}{n-i}$$

alertIdentidade de Vandermonde.

$$P_{1,1}^{(2n)} = \left(\frac{1}{4}\right)^{2n} \binom{2n}{n} \binom{2n}{n} \approx \frac{((2n)^{2n} e^{-2n} \sqrt{2\pi 2n})^2}{4^{2n} (n^n e^{-n} \sqrt{2\pi n})^4} = \frac{1}{n \cdot \pi}$$

Portanto:

$$\sum_{n=0}^{\infty} P_{1,1}^{(2n)} = \infty \quad \Rightarrow \quad \text{Todos os estados são Recorrentes}$$

Symmetric Random Walk: 1|2 dimensões (todos estados são recorrentes)
$\geq$ 3 dimensões (todos estados são transientes)

Markov Chain - Distribuição Estacionária e Periodicidade

Vetor de probabilidade $\pi^{(t)} = [\pi_1^{(t)}, \ldots, \pi_n^{(t)}]$: Matriz linha $1 \times n$ tal que $\pi_i^{(t)}$ é a probabilidade do processo estar no estado i no tempo t.

- $\pi^{(t)} = \pi^{(t-1)} \cdot P$. Ou seja, $\pi_i^{(t)} = \sum_{k=1}^{n} \pi_k^{(t-1)} \cdot P_{k,i}$
- $\Rightarrow \quad \pi^{(t)} = \pi^{(0)} \cdot P^t$, onde P é matriz de transição da Cadeia de Markov.

Distribuição limite (ou estacionária): Qualquer vetor π tq $\pi = \pi \cdot P$.

- **Intuição:** Se o vetor de prob for igual a π em algum momento, o vetor de prob do processo será sempre π.

$\sum_{i=0}^{n} \pi_i = 1$

Tempo de travessia (hitting time) T_{ij}**:** Tempo (número de passos) esperado para alcançar o estado j iniciando no estado i. $T_{ij} > 0$.

Periodicidade do estado i**:** $\max\{d : \ P_{i,i}^n = 0, \forall n$ não divisível por $d\}$. Estados de uma mesma classe tem a mesma periodicidade.

Uma cadeia é **aperiódica** se todo estado é aperiódico (periodicidade=1).

Exemplo: Random Walk tem período 2 em cada estado. Triângulo direcionado com probabilidades 1 tem periódo 3 em cada estado.

Markov Chain - Teorema Fundamental

Teorema Fundamental das Cadeias de Markov:
Toda Cadeia de Markov finita, irredutível e aperiódica satisfaz:

(a) Todos os estados são recorrentes e aperiódicos

(b) Existe uma única distribuição estacionária π tq $\pi_i > 0,\ \forall i$

(c) $f_{ii} = 1$ e $T_{ii} = 1/\pi_i$ para todo estado i

(d) Para qualquer vetor $\pi^{(0)}$ de prob inicial,

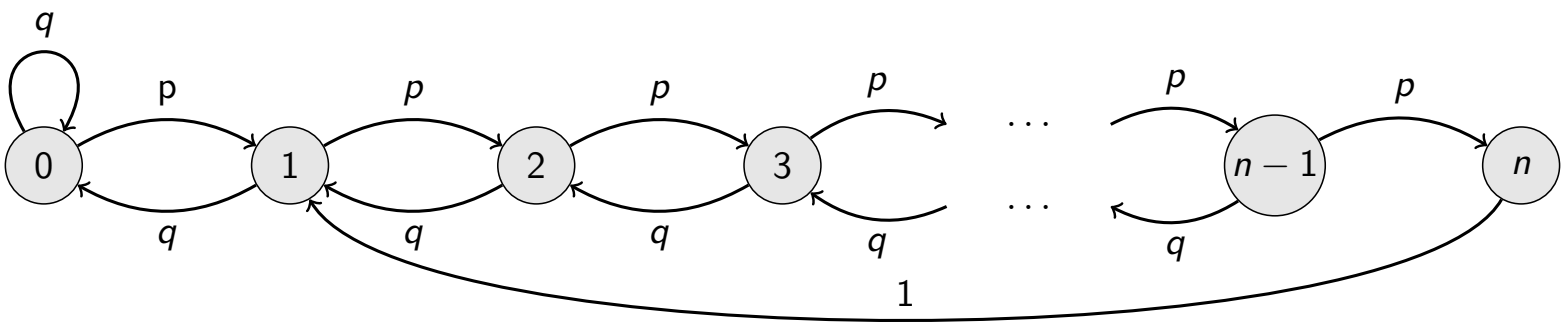
$$\lim_{t\to\infty} \pi^{(t)} = \pi \quad \text{(Convergência exponencialmente rápida)}$$

(e) Seja $N(i,t)$ o número de vezes que i é visitado em t passos.

$$\lim_{t\to\infty} \frac{N(i,t)}{t} = \pi_i$$

Markov Chain - Glória do Jogador (*Gambler's Glory*)

Probabilidade p de ganhar 1 real e prob $q = 1 - p$ de perder 1 real.
Mas, se perder tudo, consegue 1 real com um padrinho rico e recomeça.
Se ganhar n reais, dá $n - 1$ ao padrinho e recomeça.
Qual o tempo esperado para conseguir n reais começando com 1 real?



1 classe: todos os estados são recorrentes e aperiódicos.

Objetivo: Calcular o tempo esperado $T_{1,n}$ p/ ir de 1 até n.

- Método 1: Calcular $\pi_n \Rightarrow \frac{1}{\pi_n} = T_{n,n} = 1 + T_{1,n}$
- Método 2: Calcular $T_{1,n}$ diretamente em função de outros $T_{i,n}$.
- Método 3: Fazer a conta na marra: analisar cada caminho possível de 1 até n e calcular sua probabilidade.
- Método 4: Experimental p/ $T_{1,n}$. MCMC (Monte Carlo Markov Chain).
- Método 5: Experimental p/ π_n. Convergência rápida ao limite.

Markov Chain - Glória do Jogador (*Gambler's Glory*)

Método 1: Calcular $\pi_n \Rightarrow \frac{1}{\pi_n} = T_{n,n} = 1 + T_{1,n}$

Info: $\pi = \pi \cdot P$ e $\sum_{i=0}^{n} \pi_i = 1$. Portanto:

$$\begin{aligned}
\pi_n &= p \cdot \pi_{n-1} && \Rightarrow \quad \pi_{n-1} = \tfrac{1}{p} \cdot \pi_n \\
\pi_{n-1} &= p \cdot \pi_{n-2} && \Rightarrow \quad \pi_{n-2} = \tfrac{1}{p^2} \cdot \pi_n \\
\pi_{n-2} &= p \cdot \pi_{n-3} + q \cdot \pi_{n-1} && \Rightarrow \quad \pi_{n-3} = \tfrac{1}{p^3}(1 - pq) \cdot \pi_n
\end{aligned}$$

Simplificação: $p = 1/2$. **Exercício 1:** terminar p/ p qualquer.

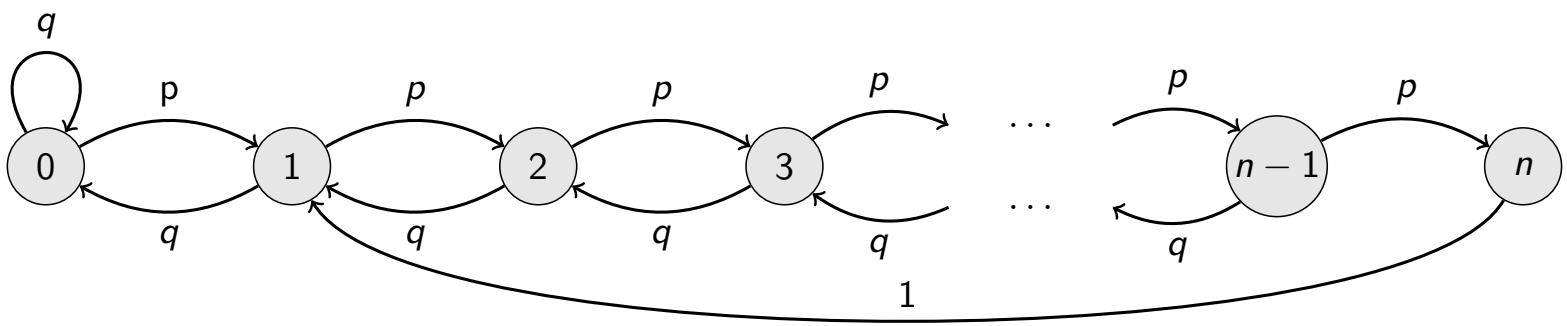
$$\begin{aligned}
\pi_n &= (1/2) \cdot \pi_{n-1} && \Rightarrow \quad \pi_{n-1} = 2 \cdot \pi_n \\
\pi_{n-1} &= (1/2) \cdot \pi_{n-2} && \Rightarrow \quad \pi_{n-2} = 4 \cdot \pi_n \\
\pi_{n-2} &= (1/2) \cdot \pi_{n-3} + (1/2) \cdot \pi_{n-1} && \Rightarrow \quad \pi_{n-3} = 6 \cdot \pi_n \\
\pi_{n-3} &= (1/2) \cdot \pi_{n-4} + (1/2) \cdot \pi_{n-2} && \Rightarrow \quad \pi_{n-4} = 8 \cdot \pi_n \\
\pi_{n-4} &= (1/2) \cdot \pi_{n-5} + (1/2) \cdot \pi_{n-3} && \Rightarrow \quad \pi_{n-5} = 10 \cdot \pi_n \\
&\dots \\
\pi_{i+1} &= (1/2) \cdot \pi_i + (1/2) \cdot \pi_{i+2} && \Rightarrow \quad \pi_i = 2(n-i) \cdot \pi_n \\
&\dots \\
\pi_2 &= (1/2) \cdot \pi_1 + (1/2) \cdot \pi_3 && \Rightarrow \quad \pi_1 = 2(n-1) \cdot \pi_n \\
\pi_0 &= (1/2)\pi_0 + (1/2)\pi_1 && \Rightarrow \quad \pi_0 = \pi_1 = 2(n-1) \cdot \pi_n
\end{aligned}$$

Markov Chain - Glória do Jogador (*Gambler's Glory*)

Método 1: Calcular $\pi_n \Rightarrow \frac{1}{\pi_n} = T_{n,n} = 1 + T_{1,n}$

Simplificação: $p = 1/2$. **Exercício 1:** terminar p/ p qualquer.

$$1 = \sum_{i=0}^{n} \pi_i = \pi_n - 2\pi_n + \sum_{i=1}^{n} 2i \cdot \pi_n = \pi_n \left(-1 + n(n+1)\right) \Rightarrow \pi_n = \frac{1}{n^2 + n - 1}$$



Portanto, do Teorema Fundamental das Cadeias de Markov:

$$T_{n,n} = \frac{1}{\pi_n} = n^2+n-1 \quad \Rightarrow \quad T_{1,n} = T_{n,n} - 1 = n^2 + n - 2 = (n+2)(n-1)$$

Markov Chain - Glória do Jogador (*Gambler's Glory*)

Método 2: Calcular $T_{1,n}$ diretamente em função de outros $T_{i,n}$.

Info: Esperança condicional: $\mathbb{E}(X) = \sum_i \mathbb{E}(X|Y=i) \cdot \mathbb{P}(Y=i)$.

Seja T_i o tempo esperado para chegar em n a partir de i, inclusive.
Formalmente, $T_i = T_{i,n}$ se $i \neq n$ e $T_n = 0$, cc.;
$T_i = 1 + p \cdot T_{i+1} + q \cdot T_{i-1}$
$T_0 = 1 + p \cdot T_1 + q \cdot T_0 \Rightarrow T_1 - T_0 = -1/p$

Objetivo: Calcular T_1. Generalizar para T_i.

$$(T_{i+1} - T_i) = \left(\frac{q}{p}\right) \cdot (T_i - T_{i-1}) - \frac{1}{p}$$

Simplificação: $p = 1/2$. **Exercício 2:** terminar p/ p qualquer.

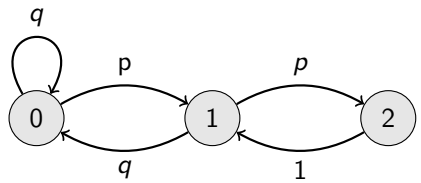
$$(T_{i+1} - T_i) = (T_1 - T_0) - 2i \Rightarrow T_{i+1} = T_i - 2(i+1) \Rightarrow T_n = T_i - 2\sum_{k=i+1}^{n} k$$

$$T_i = (n+i+1)(n-i) \quad \Rightarrow \quad T_1 = (n+2)(n-1)$$

Markov Chain - Glória do Jogador (*Gambler's Glory*)

Método 3: Fazer a conta na marra: analisar cada caminho possível de 1 a n e calcular prob. Dificuldade: num caminhos é exponencial !!

Exemplo:



$n = 2 \Rightarrow T_{1,n} = (2+2)(2-1) = 4$

1 tam 1 (12)
0 tam 2
1 tam 3 (1012)
1 tam 4 (10012)
2 tam 5 (100012, 101012)
3 tam 6 (1000012, 1001012, 1010012)
5 tam 7 (10000012, 10001012, 10010012, 10100012, 10101012)

8 tam 8 (100000012, 100001012, 100010012, 100100012, 101000012, 100101012, 101001012, 101010012)

Fibonacci $F(k-2)$ caminhos de tam. $k \geq 2$. **Exercício 3 !!**

$$T_{1,2} = 1 \cdot \frac{1}{2} + \sum_{k=2}^{\infty} k \cdot F(k-2) \cdot \left(\frac{1}{2}\right)^k = 4$$

Exercício 4 !!

desmos

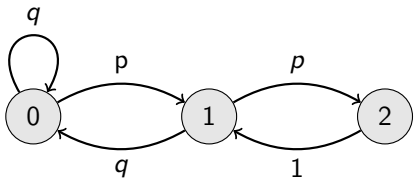
$$1 \cdot \frac{1}{2} + \sum_{n=2}^{1000} n \cdot F(n-2) \cdot \left(\frac{1}{2}\right)^n \quad = 4$$

desmos

$$F(x) = \frac{1}{\sqrt{5}} \left(\frac{\left(1+\sqrt{5}\right)}{2}\right)^x - \frac{1}{\sqrt{5}} \left(\frac{\left(1-\sqrt{5}\right)}{2}\right)^x$$

Markov Chain - Glória do Jogador (*Gambler's Glory*)

Método 4: Experimental p/ $T_{1,n}$. MCMC (Monte Carlo Markov Chain)



$$n = 2 \;\Rightarrow\; T_{1,n} = (2+2)(2-1) \;=\; 4$$

```
int main() {
    // Gerando seed pseudo-aleatória
    srand((unsigned long) time(nullptr));

    int repeticoes = 100000;
    int tempo = 0;
    int soma = 0;
    int x = 1;
    double p;

    for (int i = 0; i < repeticoes; ++i) {
        x = 1;
        tempo = 0;
        //cout << x;
        while (x!=2) {
            tempo++;
            p = (double)rand() / (double)RAND_MAX;
            if (x==0) {
                if (p < 0.5) x=1;
            } else if (x==1) {
                if (p < 0.5) x=0; else x=2;
            }
            //cout << x;
        }
        //cout << " --- " << tempo << "\n";
        soma += tempo;
    }
    cout << "Tempo medio = " << (double)soma/(double)repeticoes;
    return 0;
```

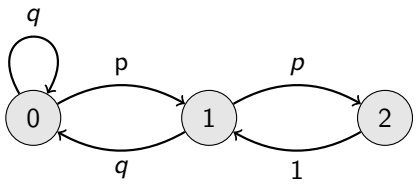
```
12 --- 1
10012 --- 4
10012 --- 4
101012 --- 5
1000101000012 --- 12
100000100012 --- 11
10012 --- 4
12 --- 1
10012 --- 4
12 --- 1
1012 --- 3
12 --- 1
101000100012 --- 11
12 --- 1
12 --- 1
12 --- 1
10012 --- 4
1000001010001000100101010001012 --- 28
12 --- 1
12 --- 1
12 --- 1
10012 --- 4
12 --- 1
10012 --- 4
10000012 --- 7
Tempo medio = 4.00514
execution time : 16.335 s
```

Exercício 5: Fazer p/ $n = 3, \ldots, 100$, listar tempos médios e comparar.

Markov Chain - Glória do Jogador (*Gambler's Glory*)

Método 5: Experimental p/ π_n. Convergência rápida ao limite.

$\frac{1}{\pi_n} = T_{n,n} = 1 + T_{1,n}$.



$n = 2 \Rightarrow T_{1,n} = (2+2)(2-1) = 4$

```
int main() {
    // Gerando seed pseudo-aleatória
    srand((unsigned long) time(nullptr));

    double pi0, pi1, pi2;
    pi0 = pi1 = pi2 = 1.0/3.0;
    cout << pi0 << "\t" << pi1 << "\t" << pi2 << "\n";

    int repeticoes = 60;
    for (int i = 0; i < repeticoes; ++i) {
        double x0 = pi0/2 + pi1/2;
        double x1 = pi0/2 + pi2;
        double x2 = pi1/2;
        pi0 = x0; pi1 = x1; pi2 = x2;
        cout << pi0 << "\t" << pi1 << "\t" << pi2 << "\n"
    }
    return 0;
```

```
0.333333    0.333333    0.333333
0.333333    0.5         0.166667
0.416667    0.333333    0.25
0.375       0.458333    0.166667
0.416667    0.354167    0.229167
0.385417    0.4375      0.177083
0.411458    0.369792    0.21875
0.390625    0.424479    0.184896
0.407552    0.380208    0.21224
0.39388     0.416016    0.190104
0.404948    0.387044    0.208008
0.395996    0.410482    0.193522
0.403239    0.39152     0.205241
0.39738     0.40686     0.19576
0.400009    0.399978    0.200014
0.399997    0.400008    0.199995
0.4         0.4         0.2
```

Exercício 6: Fazer p/ $n = 3, \ldots n = 100$, listar tempos médios.
Comparar com os valores dos slides anteriores.

Markov Chain - 2-SAT - algoritmo probabilístico polinomial

Problema k-SAT de decisão: Dada uma fórmula lógica na FNC onde cada cláusula tem no máximo k literais, existe uma atribuição às variáveis que satisfaça a fórmula?

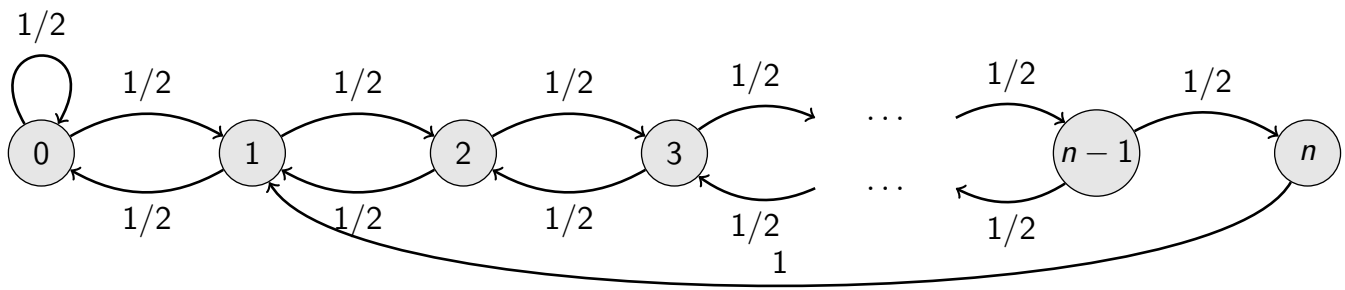
Exemplo 2-SAT: $(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2)$
Exemplo 2-SAT: $(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2})$
Literatura: k-SAT é poli $O(n^3)$ p/ $k \leq 2$, mas NP-Completo p/ $k \geq 3$.
Existem algoritmos mais rápidos p/ $k \leq 2$, mas são mais complicados.

Algoritmo probabilístico poli $O(n^2)$ p/ 2-SAT usando Cadeia de Markov.
A partir de uma atribuição qualquer às variáveis, repita $\leq 2 \cdot n(n+1)$ vezes:

- Se há cláusula insatisfeita, tome um literal uniform aleatório dela, mudando valor na atribuição. Caso contrário, retorne a atribuição. One sided error.

Análise por Cadeia de Markov: Suponha que a fórmula seja satisfatível e considere uma atribuição fixa que a satisfaz como sendo os valores "corretos" das variáveis.

Objetivo: obter n valores "corretos" ou alcançar outra atribuição que satisfaz fórmula.



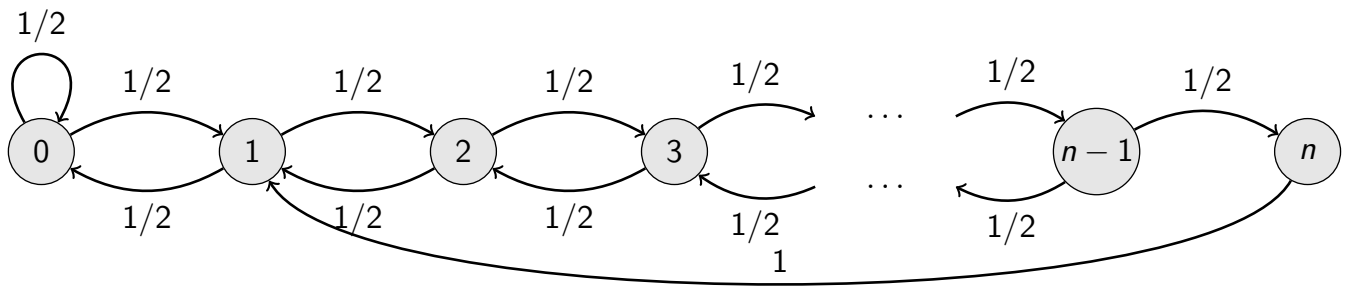
Markov Chain - 2-SAT - algoritmo probabilístico polinomial

Algoritmo probabilístico poli $O(n^2)$ p/ 2-SAT usando Cadeia de Markov.
A partir de uma atribuição qualquer às variáveis, repita $\leq 2 \cdot n(n+1)$ vezes:

- Se há cláusula insatisfeita, tome um literal uniform aleatório dela, mudando valor na atribuição. Caso contrário, retorne a atribuição. One sided error.

Análise por Cadeia de Markov: Suponha que a fórmual seja satisfatível e considere uma atribuição fixa que a satisfaz como sendo os valores "corretos" das variáveis.

Objetivo: obter n valores "corretos" ou alcançar outra atribuição que satisfaz fórmula.



Em cláusula insatisfeita, pelo menos 1 literal não está c/ valor correto. Prob $\geq 1/2$ de obter mais um valor correto e prob $\leq 1/2$ de diminuir o número de valores corretos.

Número esperado de passos: $\mathbb{E}(X) = n^2 + n - 2$. Repetindo $2n(n+1)$ vezes, temos pela Desigualdade de Markov que a probabilidade de não obter n valores corretos se a fórmula for satisfatível é $\mathbb{P}(X \geq 2 \cdot \mathbb{E}(X)) \leq 1/2$.

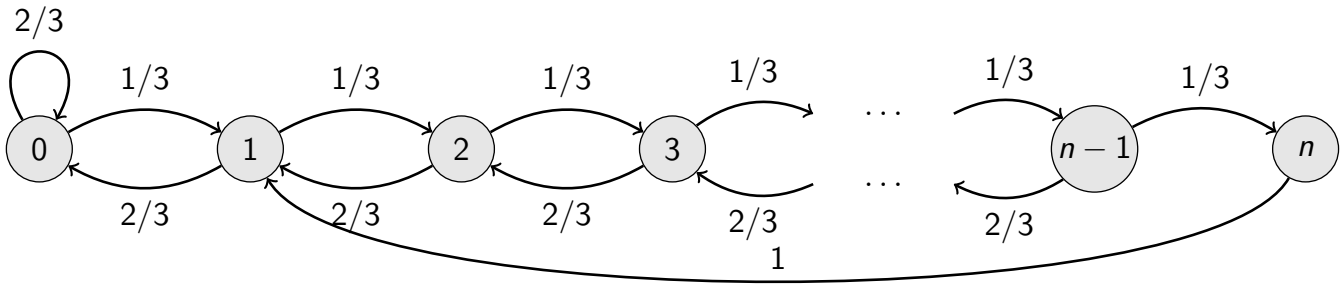
Markov Chain - 3SAT - algoritmo probab exponencial

Algoritmo probabilístico expo $O(??)$ p/ 3-SAT usando Cadeia de Markov.
A partir de uma atribuição qualquer às variáveis, repita no máximo ??? vezes:

- Se há cláusula insatisfeita, tome um literal uniform aleatório dela, mudando valor na atribuição. Caso contrário, retorne a atribuição. One sided error.

Análise por Cadeia de Markov: Suponha que a fórmual seja satisfatível e considere uma atribuição fixa que a satisfaz como sendo os valores "corretos" das variáveis.

Objetivo: obter n valores "corretos" ou alcançar outra atribuição que satisfaz fórmula.



Em cláusula insatisfeita, pelo menos 1 literal não está c/ valor correto. Prob $\geq 1/3$ de obter mais um valor correto e prob $\leq 2/3$ de diminuir o número de valores corretos.

Número esperado de passos: $\mathbb{E}(X) = ??$. Repetindo $2 \cdot \mathbb{E}(X)$ vezes, temos pela Desigualdade de Markov que a probabilidade de não obter n valores corretos se a fórmula for satisfatível é $\mathbb{P}(X \geq 2 \cdot \mathbb{E}(X)) \leq 1/2$.

Exercício !!

Markov Chain - Cadeias de Markov em Grafos

Periodicidade do estado *i*: $\max\{d : P^n_{i,i} = 0, \forall n \text{ não divisível por } d\}$.
Estados de uma mesma classe tem a mesma periodicidade.

Estado i é aperiódico se **(a)** $P_{i,i} > 0$ (laço) ou **(b)** $P^{(k)}_{i,i} > 0$ e $P^{(\ell)}_{i,i} > 0$ para k e ℓ primos entre si ($mdc(k, \ell) = 1$).

Seja G um grafo finito não-direcionado conexo. Seja M_G a cadeia de Markov induzida por G com matriz de transição com valores $P_{u,v} = 1/d(u)$ se v é vizinho de u, e 0 caso contrário, onde $d(u)$ é o grau de u em G.

$\sum_{v=1}^{n} P_{u,v} = d(u)/d(u) = 1$, onde n é o número de vértices de G.

Lema 6.3/4 (Motwani): $T_{v,v} = \frac{2m}{d(v)} = \frac{1}{\pi_v}$, se G é não-bipartido.

Prova: G é conexo não-bipartido. Logo M_G tem ciclos ímpares (e pares) p/ cada vértice. Logo, cada estado (vértice) tem periodicidade 1 e M_G é aperiódica.

$\pi_v = d(v)/2m$ é solução de $\pi = \pi \cdot P$ e $\sum_v \pi_v = 1$. Como M_G é finita, recorrente e aperiódica, a solução é única pelo Teorema Fundamental.

$$(\pi \cdot P)_v = \sum_u \pi_u \cdot P_{u,v} = \sum_{u \in Adj(v)} \left(\frac{d(u)}{2m}\right) \cdot \left(\frac{1}{d(u)}\right) = \sum_{u \in Adj(v)} \frac{1}{2m} = \frac{d(v)}{2m} = \pi_v$$

$$\sum_{v \in V(G)} \pi_v = \sum_{v \in V(G)} \frac{d(v)}{2m} = \frac{1}{2m} \sum_{v \in V(G)} d(v) = \frac{1}{2m} \cdot 2m = 1$$

Markov Chain - Cadeias de Markov em Grafos

Seja G um grafo finito não-direcionado conexo. Seja M_G a cadeia de Markov induzida por G com matriz de transição com valores $P_{u,v} = 1/d(u)$ se v é vizinho de u, e 0 caso contrário, onde $d(u)$ é o grau de u em G.

Lema 6.3/4 (Motwani): $T_{v,v} = \frac{2m}{d(v)} = \frac{1}{\pi_v}$ se G é não-bipartido.

Lema 6.6 (Motwani): $T_{u,v} + T_{v,u} = 2m \cdot R(u,v) \leq 2m \cdot d(u,v)$, onde $u \neq v$, $R(u,v)$ é a resistência efetiva entre u e v, e $d(u,v)$ é a distância entre u e v (num arestas em um menor caminho).

Para calcular a Resistência efetiva $R(u,v)$, considera-se que cada aresta do grafo representa 1 Ω (ohm) de resistência e que 1A (ampère) de corrente é introduzido em u e removido de v (ou vice-versa).

$R(u,v) \leq \ell/k$ se existem k caminhos entre u e v (disjuntos nas arestas) com tamanho no máximo ℓ (num arestas).

Markov Chain - Conectividade em Log-Space

Problema USTCON ("*Undirected s-t Connectivity*"): Dado um grafo G não-direcionado e vértices s e t, decidir se existe um caminho entre s e t.

Classe RLP (Randomized Log-space Polynomial time): Problemas de decisão que possuem algoritmo probabilístico de tempo polinomial e espaço logarítmico que sempre acerta quando retorna SIM, mas pode errar com probabilidade $1/2$ quando retorna NÃO. RLP $\subseteq$ RP

Algoritmo RLP para USTCON: O espaço não conta a memória da entrada, que é dada numa "read-only tape".

Simular passeio aleatório de tam $2n^3$ sobre (a cadeia de Markov de) G começando em s. Se encontrou t, retorne SIM. Senão, retorne NÃO.

$T_{s,t} \leq T_{s,t} + T_{t,s} \leq 2m \cdot d(s,t) \leq 2m \cdot (n-1) \leq n^3$.
Pela Desigualdade de Markov, a probabilidade de errar é $\leq \frac{n^3}{2n^3} = 1/2$.

Espaço utilizado: contador inteiro até $2n^3$ (espaço $\log_2 2n^3$), vértice atual e próximo vértice.

Técnicas Algébricas - *Fingerprint*

Técnica 1 já vista em 2 problemas:
Verificar Igualdade de polinômios e Verificar produto de matrizes.

Fingerprint 1: Para decidir se 2 objetos são iguais, ao invés de comparar os objetos por completo, tratar os objetos como funções, gerar parâmetro aleatório e comparar os valores das funções com esse parâmetro.

Fingerprint 2: Para decidir se 2 objetos são iguais, ao invés de comparar os objetos por completo, tratar os objetos como números inteiros, gerar um número primo aleatório e comparar os números nos inteiros módulo p.

Como gerar números primos aleatórios entre 2 e n: gera inteiro aleatório uniforme e testa se é primo.
Chebyshev (1848): $NPrimos(\leq n) \geq (1/3)(n/\ln n)$.
Prime Number Theorem (1896): $NPrimos(\leq n) \approx n/\ln n$.
Número esperado de buscas por um número primo: $\leq 3 \ln n$.
Tempo de checar se um inteiro é primo: $\tilde{O}((\log n)^3)$ ou $\tilde{O}((\log n)^6)$.
Tempo esperado total: $\tilde{O}((\log n)^7) = O((\log n)^{7+\varepsilon})$. Primality test (Wikipedia)

Fingerprint 1: Verificar Igualdade de Polinômios

Problema: Alice e Bob tem polinômios $A(x)$ e $B(x)$ de grau n, resp. Decidir se $A(x) = B(x)$ "*transmitindo poucos dados*". Considere que um deles não tem acesso aos coeficientes do seu polinômio.
Exemplo: $(x+1)(x-2)(x+3)(x-4) \;=\; x^4 + 7x^2 - 24$? $\quad (n = 4)$

Algoritmo: Escolher inteiro r em $\{1, \ldots, 100n\}$ aleatório uniforme. Transmitir r e $A(r)$ para Bob, que deve comparar com $B(r)$. Se iguais, retornar SIM. Caso contrário, retornar NÃO.
Bits transmitidos: $O(\log((100n)^n \cdot n)) = O(n \log n)$.

Análise: Se SIM ($A(x) = B(x)$), o algoritmo acerta.
Se NÃO ($A(x) \neq B(x)$) e $A(r) \neq B(r)$, o algoritmo acerta.
O algoritmo só erra se for NÃO ($A(x) \neq B(x)$), mas $A(r) = B(r)$.
One-sided error false biased (acerta ao dizer NÃO): Classe co-RP.

Probabilidade de errar: r deve ser raiz de $A(x) - B(x)$, que tem grau $\leq n$ e no máximo n raizes. Probabilidade de erro $\leq 1/100$

Amplificação: Repetindo k vezes com reposição: $\mathbb{P}(\text{erro}) \leq (1/100)^k$

Fingerprint 1: Verificar igualdade de matrizes

Problema: Alice e Bob tem matrizes $n \times n$ A e B, respectivamente. Decidir se $A = B$ "*transmitindo poucos dados*".

Algoritmo: Escolher vetor $r = (r_1, \ldots, r_n)$ em $\{0,1\}^n$ aleatório uniforme. Transmitir r e $A \cdot r$ para Bob, que deve comparar com $B \cdot r$. Se igual, SIM. C.C, NÃO. Bits transmitidos: $O(n)$.

Análise: Se SIM ($A = B$), o algoritmo acerta.
Se NÃO ($A \neq B$) e $A \cdot r \neq B \cdot r$, o algoritmo acerta.
O algoritmo só erra se for NÃO ($A \neq B$), mas $A \cdot r = B \cdot r$.
One-sided error false biased (acerta ao dizer NÃO): Classe co-RP.

Probabilidade de errar: Seja $D = A - B \neq 0$. Logo D tem valor $\neq 0$ (para simplificar, suponha que $d_{1,1} \neq 0$).

$$A \cdot r = B \cdot r \Rightarrow D \cdot r = 0 \Rightarrow \sum_{k=1}^{n} d_{1,k} \cdot r_k = 0 \Rightarrow r_1 = -\sum_{k=2}^{n} \frac{d_{1,k} \cdot r_k}{d_{1,1}}$$

$\mathbb{P}(r_1 \text{ ser este valor} \mid r_2, \ldots, r_n) \leq 1/2$. Logo $\mathbb{P}(\text{erro}) \leq 1/2$.

Amplificação: Repetindo k vezes com reposição: $\mathbb{P}(\text{erro}) \leq (1/2)^k$

Fingerprint 2: Verificar igualdade de palavras (*strings*)

Problema: Alice/Bob tem sequências de bits $(a_1, \ldots, a_n)$ e $(b_1, \ldots, b_n)$. Verificar se são iguais com "*transmissão de poucos dados*".

Algoritmo: Seja $a = \sum_{i=1}^{n} a_i \cdot 2^{i-1}$ e $b = \sum_{i=1}^{n} b_i \cdot 2^{i-1}$.
Seja $F_p(x) = (x \mod p)$, onde p é um primo aleat entre 2 e $m = tn \ln tn$ para um $t = O(n)$. Alice transmite p e $F_p(a)$ para Bob, que deve comparar com $F_p(b)$. Número de bits transmitidos: $O(\log p) = O(\log n)$.

$F_p(a) \neq F_p(b) \;\Rightarrow\; a \neq b$
$F_p(a) = F_p(b) \;\Rightarrow\; (a = b)$ ou p é divisor de $|a - b| \leq 2^n$

$|a - b|$ tem $\leq n$ divisores primos. Logo (Chebyshev: $NPrimos_{\leq m} \geq m/3 \ln m$)

$$\mathbb{P}(F_p(a) = F_p(b) \mid a \neq b) \leq \frac{n}{NPrimos_{\leq m}} \leq \frac{n}{m/3 \ln m} \leq \frac{3}{t} \cdot \left(1 + \frac{\ln \ln tn}{\ln tn}\right) \leq \frac{5}{t}$$

Fingerprint 2: Pattern Matching

Problema: Dadas palavras $X = x_1 \dots x_n$ e $Y = y_1 \dots y_{k \leq n}$ num mesmo alfabeto fixo Σ, checar se o padrão Y ocorre contiguamente em X.

Força bruta: Tempo $O(n \cdot k)$. **Knuth-Morris-Pratt:** Tempo $O(n+k) = O(n)$

Seja $X(j) = x_j x_{j+1} \dots x_{j+k-1}$ para $j \in \{1, \dots, n-k+1\}$. Para simplificar a explicação, seja $\Sigma = \{0, 1\}$. Considere $X(j)$ e Y como inteiros com k bits.
Objetivo: checar se existe j tal que $X(j) = Y$.

Fato: $X_{j+1} = 2 \cdot (X(j) - 2^{k-1} x_j) \ + \ x_{j+k}$. Dado p, seja $F_p(x) = (x \mod p)$.
Logo: $F_p(X_{j+1}) = 2 \cdot (F_p(X(j)) - 2^{k-1} x_j) \ + \ x_{j+k} \mod p$.

Alg. Monte Carlo $O(n+k)$: Sorteie p primo aleat entre 2 e $m = n^2 k \ln n^2 k$.
Calcule todos os valores $F_p(X(j))$ para $j \in \{1, \dots, n-k+1\}$.
Se algum for igual $F_p(Y)$, retorne SIM. Caso contrário, retorne NÃO.

Para um certo j fixo: $F_p(Y) \neq F_p(X(j)) \ \Rightarrow$ sem padrão Y começando em j
$F_p(Y) = F_p(X(j)) \ \Rightarrow$ (com padrão) ou (p é divisor de $|Y - X(j)| \leq 2^k$)

$|Y - X(j)|$ tem $\leq k$ divisores primos. Logo (Chebyshev: $NPrimos_{\leq m} \geq m/3 \ln m$)

$$\mathbb{P}(\ F_p(Y) = F_p(X(j)) \ \mid \ Y \neq X(j)\ ) \leq \frac{k}{NPrimos_{\leq m}} \leq \frac{k}{m/3 \ln m} \leq \frac{5}{n^2}$$

Prob $\leq \frac{5}{n}$ de $\exists j$ com $F_p(Y) = F_p(X(j))$ apesar de Y não ser padrão.

Fingerprint 2: Pattern Matching

Alg. Monte Carlo $O(n+k)$: Sorteie p primo aleat entre 2 e $m = n^2 k \ln n^2 k$.
Calcule todos os valores $F_p(X(j))$ para $j \in \{1, \ldots, n-k+1\}$.
Se algum for igual $F_p(Y)$, retorne SIM. Caso contrário, retorne NÃO.
Probabilidade de erro $\leq 5/n$.

Alg. Las Vegas $O(n+k)$: Roda o algoritmo Monte Carlo e, se apontar um padrão $F_p(X(j)) = F_p(Y)$, comparar as strings $X(j)$ e Y em tempo $O(k)$.
Se igual, retorna SIM. Caso contrário, reinicia tudo com Força Bruta.
Probabilidade de erro $= 0$.

Tempo esperado do Algoritmo Las Vegas:

$$\leq \left(1 - \frac{5}{n}\right) \cdot O(n+k) \; + \; \left(\frac{5}{n}\right) \cdot O(n \cdot k) \; = \; O(n+k)$$

Teste de Primalidade de Fermat

Problema: Dado um número inteiro n, verificar se é primo.

Força bruta: Testar divisibilidade de n com todo inteiro k de 2 até $\sqrt{n}$.

Teste de Primalidade de Fermat: Tempo $\tilde{O}((\log n)^2)$ soft-O notation

$\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. $\mathbb{Z}_n^{cp} \subseteq \mathbb{Z}_n$: coprimos de n; $mdc(x, n) = 1$.

Fermat's Little Theorem: n primo $\Rightarrow \forall a \in \mathbb{Z}_n^{cp} : \; a^{n-1} \equiv 1 \mod n$.

n composto $\Leftarrow \exists a \in \mathbb{Z}_n^{*} : \; a^{n-1} \not\equiv 1 \mod n$.

Lema 14.28 (Motwani): Se n é composto não-Carmichael, então pelo menos metade dos elementos $a \in \mathbb{Z}_n^{cp}$ satisfazem $a^{n-1} \not\equiv 1 \mod n$

Definição: n é Carmichael se é composto e $a^{n-1} \equiv 1 \mod n$, $\forall a \in \mathbb{Z}_n^{cp}$.

Números de Carmichael são raros: só 646 entre 1 e 10^9.

561 $(3 \cdot 11 \cdot 17)$, 1105 $(5 \cdot 13 \cdot 17)$, 1729 $(7 \cdot 13 \cdot 19)$, 2465 $(5 \cdot 17 \cdot 29)$, 2821 $(7 \cdot 13 \cdot 31)$, 6601 $(7 \cdot 23 \cdot 41)$, 8911 $(7 \cdot 19 \cdot 67)$.

Alg. Fermat: Sorteia $2 \leq a \leq n-2$. Se $mdc(a, n) \neq 1$, retorne NÃO.
Testa se $a^{n-1} \equiv 1 \mod n$. Se SIM, retorne SIM. C.C., retorne NÃO.

n primo $\Rightarrow$ Fermat acerta 100%.

n composto não-Carmichael $\Rightarrow$ Fermat acerta com prob $\geq 1/2$.

n composto Carmichael $\Rightarrow$ Fermat acerta se $mdc(a, n) \geq 2$.

Teste de Primalidade de Fermat

Problema: Dado um número inteiro n, verificar se é primo.

Teste de Primalidade de Fermat: Tempo $\tilde{O}((\log n)^2)$ soft-O notation
$\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$. $\mathbb{Z}_n^{cp} \subseteq \mathbb{Z}_n$: coprimos de n; $mdc(x, n) = 1$.

Alg. Fermat: Sorteia $2 \leq a \leq n-2$. Se $mdc(a, n) \neq 1$, retorne NÃO.
Testa se $a^{n-1} \equiv 1 \mod n$. Se SIM, retorne SIM. C.C., retorne NÃO.

Algoritmo de Euclides para calcular $mdc(a, n)$. Tempo $O(\log n)$.

Algoritmo ExponenciaçãoBinária (inteiros *base*, *exp*, *n*)

```
result = 1;    base ← base % n
enquanto (exp > 0)
    se (exp & 1): result ← (result · base) % n
    base ← (base · base) % n
    exp ← exp >> 1
retorne result
```

Binary Exponentiation (Wikipedia)

Teste de Primalidade de Miller-Rabin

Problema: Dado um número inteiro n, verificar se é primo.

n ímpar $\Rightarrow n-1 = 2^s \cdot d$ para $s \geq 1$ e d ímpar.

$$
\begin{aligned}
a^{n-1} \equiv 1 \bmod n &\iff a^{2^s d} - 1 \equiv 0 \bmod n \\
&\iff (a^{2^{s-1} d} + 1)(a^{2^{s-1} d} - 1) \equiv 0 \bmod n \\
&\iff (a^{2^{s-1} d} + 1)(a^{2^{s-2} d} + 1)(a^{2^{s-2} d} - 1) \equiv 0 \bmod n \\
&\;\;\vdots \\
&\iff (a^{2^{s-1} d} + 1)(a^{2^{s-2} d} + 1) \cdots (a^{d} + 1)(a^{d} - 1) \equiv 0 \bmod n
\end{aligned}
$$

Se n é ímpar, um desses fatores é múltiplo de n.

Alg. Miller-Rabin: Sorteia $2 \leq a \leq n-2$. Calcular d e s.
Testa se $a^d \equiv 1 \mod n$ ou $a^{2^r d} \equiv -1 \mod n$ para algum $0 \leq r < s$.
Se SIM, retorne SIM. C.C., retorne NÃO.

n primo $\Rightarrow$ Miller-Rabin acerta 100%.
n composto $\Rightarrow$ Miller-Rabin acerta com prob. $\geq 3/4$.
Primality é co-RP e Compositeness é RP, ambos c/ tempo $\tilde{O}((\log n)^3)$.

Obs: Teste de Primalidade AKS: determinístico tempo poli $\tilde{O}((\log n)^6)$.
Artigo famoso "PRIMES is in P", 2002. Prêmio Gödel-2006.

Teste de Primalidade de Miller-Rabin

n ímpar $\Rightarrow n-1 = 2^s \cdot d$ para $s \geq 1$ e d ímpar.

$$
\begin{aligned}
a^{n-1} \equiv 1 \bmod n \quad &\iff a^{2^s d} - 1 \equiv 0 \bmod n \\
&\iff (a^{2^{s-1} d} + 1)(a^{2^{s-1} d} - 1) \equiv 0 \bmod n \\
&\iff (a^{2^{s-1} d} + 1)(a^{2^{s-2} d} + 1)(a^{2^{s-2} d} - 1) \equiv 0 \bmod n \\
&\;\;\vdots \\
&\iff (a^{2^{s-1} d} + 1)(a^{2^{s-2} d} + 1) \cdots (a^{d} + 1)(a^{d} - 1) \equiv 0 \bmod n
\end{aligned}
$$

Algoritmo Cálculo-s-d (inteiro n)

$s = 0; \quad d = n - 1$
enquanto $((d \& 1) = 0)$
$d \leftarrow d >> 1; \quad s \leftarrow s + 1$

Algoritmo Composto (inteiros n, a, d, s)

$R \leftarrow$ ExponenciaçãoBinária(a, d, n)
se $(R = 1)$ **ou** $(R = n - 1)$: **retorne** falso
para $t = 1$ **até** $s - 1$
$R \leftarrow (R \cdot R) \,\%\, n$
se $(R = n - 1)$: **retorne** falso
retorne verdadeiro

Teste de Primalidade de Miller-Rabin (Determístico)

Problema: Dado um número inteiro n, verificar se é primo.

Alg. Miller-Rabin: Sorteia $2 \leq a \leq n-2$. Calcular d e s.
Testa se $a^d \equiv 1 \mod n$ ou $a^{2^r d} \equiv -1 \mod n$ para algum $0 \leq r < s$.
Se SIM, retorne SIM. C.C., retorne NÃO.

n primo $\Rightarrow$ Miller-Rabin acerta 100%.
n composto $\Rightarrow$ Miller-Rabin acerta com prob. $\geq 3/4$.
Primality é co-RP e Compositeness é RP, ambos c/ tempo $\tilde{O}((\log n)^3)$.

Versão Determinística (32 bits): $a = 2, 3, 5, 7, 11$ (5 primeiros primos).
Versão Determinística (64 bits): $a = 2, \ldots, 37$ (12 primeiros primos).

Obs: Teste de Primalidade AKS: determinístico tempo poli $\tilde{O}((\log n)^6)$.
Artigo famoso "PRIMES is in P", 2002. Prêmio Gödel-2006.

Exercício: Implementar algoritmos de teste de primalidade de Fermat e Miller-Rabin e calcular experimentalmente (com pelo menos 100000 execuções para n **composto** aleatório uniforme entre 100 e 100000) as probabilidades de erro para um sorteio apenas do inteiro $2 \leq a \leq n-2$. e comparar com $1/2$ e $1/4$, respectivamente.