Ministério da Educação
Universidade Federal do Ceará
Pró-Reitoria de Pesquisa e Pós-Graduação

FORMULÁRIO PARA CRIAÇÃO DE DISCIPLINAS

1. Identificação do Curso:		
1.1 Curso:	Ciência da Computação	
1.2 Código:	P37 (mestrado) P63 (doutorado)	
2. Modalidades:		
Mestrado (X)	Doutorado (X)	
3. Turno(s)		
Diurno (X)	Noturno ( )	
4. Departamento		
Departamento de Computação		
5. Identificação da Disciplina:		
Nome :	Algoritmos Probabilísticos	
Código:		
Carga Horária:	64h	
Nº de Créditos:	4	
Optativa:	Sim (X)	Não ( )
Obrigatória:	Sim ( )	Não (X)
6. Pré-Requisitos:		
Construção e Análise de Algoritmos Probabilidade e Processos Estocásticos		
7. Professor Responsável:		
Carlos Eduardo Fish de Brito Rafael Castro de Andrade Rudini Menezes Sampaio Victor Almeida Campos		

8. JUSTIFICATIVA

Dado um problema NP-Difícil como Satisfatibilidade, Caixeiro Viajante ou Coloração mínima de vértices, sabe-se que a obtenção de algoritmos exatos em tempo polinomial implicaria P=NP. Então, ao invés de algoritmos exatos, uma alternativa é buscar algoritmos aproximativos ou algoritmos probabilísticos. Um algoritmo probabilístico é um algoritmo que usa internamente algum tipo de geração aleatória e cuja resposta ou tempo de execução podem depender desse fator aleatório. Se o tempo de execução, mas não a resposta, depender do processo aleatório, dizemos que o algoritmo é um algoritmo *Las Vegas*. Caso contrário, dizemos que o algoritmo é um algoritmo *Monte Carlo*.

Um exemplo conhecido de algoritmo *Las Vegas* é o algoritmo Quick-Sort aleatório. O algoritmo Quick-Sort normal escolhe o elemento pivô de forma determinística (por exemplo, tomando o último elemento) e por isso, o seu tempo de execução no pior caso é $O(n^2)$ (quando o vetor é crescente). Para evitar tal execução ruim, basta tomar o pivô aleatoriamente, o que leva a execuções da ordem $O(n \log n)$. Um exemplo conhecido de algoritmo *Monte Carlo* é o algoritmo de Johnson para MaxSAT, onde escolhe-se V ou F para uma variável com probabilidade 50%. Existem vários exemplos de algoritmos probabilísticos eficientes para problemas NP-Difíceis.

Para analisar um algoritmo probabilístico, é interessante também saber como se comportam suas estruturas de dados em um processo aleatório. Por exemplo, é importante saber a altura máxima que uma árvore usada em um algoritmo pode alcançar. Outra questão é saber como funcionam os algoritmos dos geradores pseudo-aleatórios.

Uma vez obtido um algoritmo probabilístico, em alguns casos é possível "*desaleatorizar*" o algoritmo e obter um algoritmo determinístico com propriedades similares. Os exemplos do Quick-Sort e MaxSAT possuem versões desaleatorizadas.

Outra questão de suma importância é fazer uma análise probabilística (ou de caso médio) de algoritmos determinísticos, ou seja, considerando entradas aleatórias. São conhecidos diversos algoritmos cujo tempo de pior caso é exponencial, mas, na grande maioria dos casos, sua execução é em tempo polinomial. Como exemplos conhecidos, temos o Quick-Sort normal e um algoritmo simples para obtenção do k-ésimo mínimo elemento de um vetor.

Para se elaborar um algoritmo probabilístico, bem como fazer uma análise probabilística de um algoritmo determinístico, é imprescindível o conhecimento e aplicação de certas técnicas e ferramentas de probabilidade, como amplificação, concentração de medida, desigualdades de grandes desvios, martingais, método probabilístico, entre outros.

Outra questão de suma importância é a classificação de problemas com relação a obtenção de algoritmos probabilísticos eficientes. Dizemos que um problema de decisão é da classe BPP se existe um algoritmo probabilístico polinomial que "*acerta*" a resposta com probabilidade 2/3. Ou seja, se a entrada satisfaz o problema, o algoritmo responde SIM com probabilidade 2/3. Caso contrário, o algoritmo responde NÃO com probabilidade 2/3. Existem outras classes interessantes de complexidade de problemas relacionadas a aleatoriedade, como as classes, RP e ZPP. Existem várias questões em aberto com relação ao relacionamento entre essas classes . Por exemplo, o grande problema dessa área é saber se P=BPP. Sabe-se que essa questão é equivalente a questão P=NP.

9. OBJETIVOS

Introduzir os conceitos e técnicas para o desenvolvimento de algoritmos probabilísticos, para a análise probabilística de algoritmos e para classificação de problemas com relação a existência de algoritmos probabilísticos polinomiais.

10. EMENTA

Notação básica, Exemplos básicos, Análise probabilística de algoritmos, Ferramentas de probabilidade, Desigualdades básicas, Desigualdades de grandes desvios, Martingais, Método probabilístico, Cadeias de Markov, Método de Monte-Carlo, Construção de algoritmos probabilísticos, Aplicações para problemas NP-Difíceis, Análise de estruturas de dados em processos aleatórios, Geradores pesudo-aleatórios, Classificação de problemas.

11. PROGRAMA DA DISCIPLINA

1. Introdução: motivação; exemplos básicos: amplificação probabilística; Quick-Sort, MaxSAT, Min-Cut, k-ésimo mínimo; outros exemplos.
2. Técnicas probabilísticas: Desigualdades básicas (Markov, Chebyschev), Desigualdades de grandes desvios (Chernoff, Martingais), Método probabilístico, Cadeias de Markov.
3. Análise probabilística de algoritmos; exemplos.
4. Construção e Análise de Algoritmos Probabilísticos; exemplos de problemas (Caixeiro viajante geométrico, programação linear, ...).

5. Comportamento de estruturas de dados em processos aleatórios, Grafos aleatórios. Geradores pseudo-aleatórios.

6. Classificação de problemas com relação a existência de algoritmos probabilísticos; Classes BPP, RP, ZPP; Testabilidade de Propriedades e algoritmos sub-lineares.

12. FORMA DE AVALIAÇÃO

(*) Avaliações parciais
(*) Listas de exercícios

A nota final será estabelecida de acordo com o regimento atual da universidade.

13. BIBLIOGRAFIA BÁSICA

(*) **M. Mitzenmacher, E. Upfal,** Probability and computing: randomized algorithms and probabilistic analysis**, Cambridge, 2005.**

(*) **R. Motwani, P. Raghavan**, Randomized algorithms**, Cambridge, 1995.**

14. BIBLIOGRAFIA COMPLEMENTAR

(*) **D. Dubhashi, A. Panconesi**, Concentration of measure for the analysis of randomized algorithms**, Cambridge, 2009.**

(*) **Sheldon Ross**, Probability Models for Computer Science**, Academic Press, 2002.**

Aprovado em Reunião do Colegiado da Coordenação do Curso em:

Fortaleza, _____/_____/_____

____

Coordenador(a)

Aprovado em Reunião do Conselho do Departamento em:

Fortaleza, _____/_____/_____

____

Chefe do Departamento

Aprovado em Reunião do Conselho de Centro/Faculdade em:

Fortaleza, _____/_____/_____

____

Diretor(a)

Aprovado em Reunião do Conselho de Ensino, Pesquisa e Extensão em:

Fortaleza, _____/_____/_____

____

Pró-Reitor(a) de Pesquisa e Pós-Graduação