

**Universidade Federal de Minas Gerais**  
**Programa de Pós-Graduação em Ciência da Computação**  
**Exame de Qualificação 1º Estágio**  
**1º Semestre de 2014**

**Área: Teoria: Estrutura de Dados, Projeto e Análise de Algoritmos, Técnicas de Programação, Pesquisa e Ordenação**

Em 11/03/2014, 10:00 horas

Prova individual sem consulta com duração de 2 horas

**Observações:**

1. A prova deve ser resolvida no próprio caderno de questões.
2. As questões desta prova estão nas páginas seguintes, as quais estão numeradas de 1 a 12.
3. Faz parte da prova a interpretação das questões. Caso você ache que falta algum detalhe nos enunciados ou nos esclarecimentos, você deverá fazer as suposições que achar necessárias e escrever essas suposições juntamente com as respostas.
4. **Todas** as respostas devem ser justificadas.
5. Somente serão corrigidas respostas legíveis.
6. Não se esqueça de escrever seu nome abaixo.

Desejamos a você uma boa prova!

A COPEQ

---

**Atenção:** Esta prova contém um total de 6 (seis) questões, das quais você deve fazer 4 (quatro). Marque abaixo as questões que devem ser consideradas para avaliação:

1    2    3    4    5    6    (selecione até quatro)

Nome: \_\_\_\_\_

Assinatura: \_\_\_\_\_

## Questão 1

Prove que as afirmativas abaixo são verdadeiras ou falsas, usando para isso as definições das notações assintóticas e/ou contra exemplos. Considere que as funções  $f(n)$  e  $g(n)$  são assintoticamente positivas.

- a) Se  $f(n) = O(g(n))$  e  $g(n) = O(f(n))$  então  $f(n) = \Theta(g(n))$ .
- b)  $f(n) = \Theta(f(n/2))$ .
- c)  $2^{n-1} = \Omega(2^n)$ .
- d) Se  $g(n) = \omega(f(n))$  então  $f(n) + g(n) = \Theta(g(n))$ .



## Questão 2

Uma fila de prioridades é um Tipo Abstrato de Dados (TAD) normalmente utilizado quando se deseja obter o maior elemento de um conjunto. Operações comuns nesse TAD são: construir uma fila de prioridades a partir de um conjunto de  $n$  elementos, obter o maior elemento, e inserir um novo elemento no conjunto.

- a) Uma árvore binária de pesquisa pode ser utilizada na implementação de uma fila de prioridades. Discuta a sua eficiência para as três operações mencionadas acima em comparação com i) um vetor não-ordenado ii) um vetor ordenado, iii) um heap binário.
- b) Filas de Prioridade podem ser usadas na implementação de algoritmos de ordenação. Dois exemplos são o *Algoritmo da Seleção* (quando um vetor não ordenado é utilizado) e o *Heapsort* (quando um heap binário é utilizado). Descreva em linhas gerais como seria um algoritmo de ordenação baseado em uma fila de prioridades implementada utilizando-se uma árvore binária de pesquisa. Qual seria a sua ordem complexidade?



### Questão 3

Escreva um algoritmo de complexidade  $\Theta(n \log n)$  que, dado um conjunto  $S$  com  $n$  números inteiros e um outro número inteiro  $x$ , determine se existem ou não dois elementos em  $S$  cuja soma seja  $x$ . Mostre que o seu algoritmo tem a ordem complexidade requerida.



## Questão 4

Considere o problema de encontrar uma Árvore Geradora Mínima (AGM) de um grafo conectado ponderado  $G$ .

- a) Apresente o pseudo-código dos algoritmos de *Prim* e *Kruskal* para computação da AGM de  $G$  e explique a diferença entre os dois algoritmos.
- b) Escreva um algoritmo que ache a segunda AGM de  $G$  com  $n$  arestas, ou seja, uma árvore geradora de  $G$  com o segundo menor peso.





## Questão 5

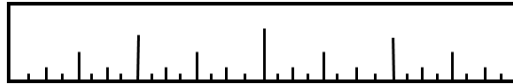
Responda as seguintes perguntas sobre problemas P, NP e NP-Completo:

- a) O problema de saber se um grafo  $G$  possui  $N$  colorações diferentes, cada uma com  $K$  cores no máximo, é NP-completo? (Uma coloração é válida se dois vértices adjacentes receberem cores diferentes)
- b) O problema de saber se um grafo  $G$  possui um clique de tamanho  $K$  é NP-Completo?
- c) O problema de encontrar o caminho mínimo entre todos os pares de vértices de um grafo  $G$  é NP?
- d) O problema de saber se existe um caminho de valor menor que  $K$  entre dois vértices de um grafo  $G$  é NP-Completo?



## Questão 6

Considere o problema de fazer marcações em uma régua com  $n$  centímetros (para simplificar considere  $n$  uma potência de 2, ou seja  $n = 2^k$ ). As marcações devem ser feitas da seguinte forma: a posição central da régua deve ter uma marca maior (de tamanho  $k$ ), seguido de marcas a cada  $1/4$  da régua ( $k - 1$ ), depois  $1/8$ , e assim por diante até que a última marca tenha o tamanho mínimo igual a 1, conforme o exemplo da figura abaixo:



- Desenvolva um algoritmo recursivo usando o paradigma *Dividir-para-Conquistar* para resolver esse problema. Considere que existe uma função  $\text{marca}(x, h)$  de custo constante que faz uma marca de tamanho  $h$  na posição  $x$  da régua.
- Qual é a complexidade do seu algoritmo? Para isso, determine e resolva a sua equação de recorrência.
- Qual é a sequência das chamadas recursivas que serão executadas? Qual o número total de chamadas? Qual o tamanho máximo da pilha de execução?

