Exame de Qualificação

04/03/2015

Nome:

Dicas gerais: *Leia todas as questões antes de começar; sempre justifique a sua resposta. A avaliação levará em consideração a abrangência e a profundidade demonstradas nas soluções apresentadas. Nas questões algorítmicas espera-se uma prova da corretude do algoritmo e uma análise do tempo em função do tamanho da entrada. As respostas podem ser breves e focadas na questão. Por favor apresente pseudo-código, caso seja necessário, nunca código explicito.*

Questão 1 (2.5 pontos) Um professor gostaria de automatizar a correção dos exercícios de programação Python dos alunos, construindo um programa que implemente a função *PROG-CORRETO?*, definida assim:

Entrada: Dois programas escritos em Python, um representando a solução (*sol*) e outro o programa submetido pelo aluno (*sub*).

Saída: *Verdadeiro*, se o programa *sub* implementa a mesma função que o programa *sol*, ou seja, se para todas as entradas para as quais *sol* produz um valor de saída, *sub* produz o mesmo valor, e para todas as entradas para as quais *sol* não produz saída, *sub* também não produz saída. A saída deve ser *Falso* caso contrário.

Responda às seguintes questões:

a) Explique o que é uma **"função computável"**.

b) A função *PROG-CORRETO?* é computável? Prove.

c) Transforme a definição de *PROG-CORRETO?* em *PROG-CORRETO?mod* da seguinte forma:

- Se sua resposta na questão b) foi SIM, *PROG-CORRETO?mod* deve ser uma função mais geral que *PROG-CORRETO?* que não seja computável.
- Se sua resposta na questão b) foi NÃO, *PROG-CORRETO?mod* deve ser uma função mais restrita que *PROG-CORRETO?* que seja computável.

Justifique por que *PROG-CORRETO?mod* é computável/não computável.

Questão 2 (1 ponto) Disserte sobre as principais classes de complexidade e explique a importância de se demonstrar a classe de complexidade de um problema.

Questão 3 (1.5 pontos) Diga se as afirmações a seguir são verdadeiras ou falsas e justifique suas respostas.

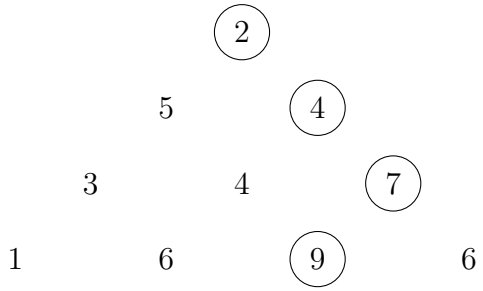
1. Se um problema pertence à classe **P**, sempre existe um algoritmo com complexidade $O(2^n)$ que o resolve.

2. Se $\mathbf{P} = \mathbf{NP}$ então a lógica de primeira ordem seria decidível.

3. Se L é **NP**-difícil, L é polinomialmente redutível a L' e L' pertence a **P** então $\mathbf{P} \neq \mathbf{NP}$.

Questão 4 (1 pontos) Você tem que fazer $n \geq 1$ panquecas usando uma frigideira com uma capacidade de no máximo duas panquecas no mesmo tempo. Cada panqueca tem que ser assado nos dois lados; assar um lado de uma panqueca precisa 1 minuto, independente se uma ou duas panquecas são assadas no mesmo tempo. Projete um algoritmo para que completa essa tarefa no tempo mínimo. Qual o tempo mínimo em função de n?

Questão 5 (2 pontos) Alguns números positivos são organizados num triângulo como, por exemplo,



Projete um algoritmo (mais eficiente que uma busca exaustiva) que encontra a maior soma numa descida da ponta superior até a base via uma sequência de números adjacentes, uma por nível.

Questão 6 (2 pontos) Na análise de um algoritmo de divisão e conquista chegamos na recorrência

$$T(n) = \begin{cases} 7T(\lfloor n/2 \rfloor) + 3n & \text{para } n > 0 \\ 0 & \text{caso contrário} \end{cases}$$

Resolve a recorrência e dá uma fórmula fechada para o crescimento assintótico de T (i.e. $T(n) = O(f(n))$ para um $f(n)$.)