PUC-Rio | Departamento de Informática
Data: 15 de dezembro de 1999 | Período: 1999.2

Projeto e Análise de Algoritmos

Prova Final

Questão 1 (3.0 pontos): Assinale *verdadeiro* (V) ou *falso* (F) para cada afirmativa abaixo e justifique (através de demonstração ou contra-exemplo, como for apropriado).

1. ( ): $\log 3^n = O(\log 2^n)$
2. ( ): $f(n) = O(g(n))$ e $g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$
3. ( ): $f(n) = O(g(n)) \Rightarrow 2^{f(n)} = O(2^{g(n)})$
4. ( ): Sejam $f(n)$ e $g(n)$ duas funções definidas sobre os números inteiros. Então, $f(n) = O(g(n))$ ou $g(n) = O(f(n))$.
5. ( ): A existência de um algoritmo de complexidade $O(\sqrt{n})$ para identificar se um número inteiro n é primo permite concluir que este problema pertence à classe P.
6. ( ): A existência de um algoritmo de complexidade $O(n)$ para identificar se um número inteiro n é primo permite concluir que este problema pertence à classe P.

Questão 2 (2.0 pontos): Seja $a = (a_1, a_2, \cdots, a_n)$ um vetor que armazena n números reais. Especifique o algoritmo de menor complexidade possível para relacionar a quantidade de ocorrências de cada número. Analise a complexidade deste algoritmo.

Questão 3 (2.0 pontos): Dado um número inteiro a, especifique um algoritmo de exponenciação com a menor complexidade possível para calcular a^n. Analise a complexidade deste algoritmo (é possível conseguir um algoritmo melhor do que aquele trivial, cuja complexidade é linear).

Questão 4 (3.0 pontos): Seja $G = [V, E]$ um grafo conexo não-orientado, onde $V = \{1, 2, \ldots, n\}$ é o conjunto de nós e E é o conjunto de arestas. Define-se um vértice de articulação de G como sendo qualquer vértice $v \in V$ cuja eliminação transforma G em um grafo não-conexo. Diz-se que dois nós $p, q \in V$ pertencem à mesma componente biconexa de G caso existam pelo menos dois caminhos nós-disjuntos (isto é, não possuem nós comuns) conectando p e q em G.

- Especifique um algoritmo de baixa complexidade para determinar os vértices de articulação de G e analise sua complexidade.
- Especifique um algoritmo de baixa complexidade para determinar as componentes biconexas de G e analise sua complexidade.

(continua no verso)

Observações:

- Prova sem consulta de qualquer espécie.
- Apresente os algoritmos solicitados sob forma de descrições algorítmicas e formais (pseudo-linguagens, por exemplo), não como códigos em determinada linguagem de programação nem usando frases imprecisas ou ambíguas em português ou outra língua.
- A correção da prova levará em consideração a exatidão, o rigor e o formalismo na especificação dos algoritmos.
- Justifique sua afirmações e as idéias empregadas no projeto dos algoritmos.
- Defina a notação utilizada e explique os conceitos empregados.
- Justifique todos seus cálculos e afirmações.