Exame Preliminar para o Doutorado
Prova de Análise de Algoritmos

IME–USP, 14 de agosto de 2009
das 14:00 às 18:30 horas

14 de agosto de 2009

Instruções:

(i) Você pode resolver todas as questões.
(ii) A banca considerará questões cujos valores somem até 10 pontos de modo que a nota seja máxima; isto será feito em tempo $O(1)$.
(iii) Mencione os teoremas e propriedades usados para justificar suas afirmações.
(iv) Algoritmos devem ser escritos em pseudo-código bem comentado, ou informalmente, mas com precisão.

Questão 1 [1 ponto]

O Professor Htunk descobriu um algoritmo de ordenação baseado em comparações que consome tempo $O(n \log \sqrt{n})$. Considerando-se o limitante inferior para o tempo de ordenação, como isso é possível? E se ele tivesse dito que o número de comparações efetuada pelo algoritmo é não mais que $n \log \sqrt{n}$, faria diferença?

Questão 2 [1 ponto]

Descreva um algoritmo que, dados n inteiros com valores entre 1 e k, pré-processe esses inteiros e então (descontado o tempo do pré-processamento) responda perguntas da forma "dados a, b, quantos dos n inteiros estão no intervalo $[a\,,\,b]$" em tempo $O(1)$. O pré-processamento deve ser feito em tempo $O(n+k)$.

Questão 3 [1 ponto]

Descreva uma implementação de um algoritmo *Heap-Remova*(A, i, n) que remove o item do nó i de um heap $A[1\,..\,n]$. Seu procedimento deve consumir tempo $O(\log n)$.

Questão 4 [2 pontos]

A classe `VObsc` tem como métodos um inteiro constante `BIG` e uma função `Valor`: $\mathbb{N} \to \mathbb{N}$. O que é garantido é que

- existe um inteiro n tal que `Valor` é crescente no intervalo $[0\,,\,n]$,
- `Valor`$(n) <$ `BIG`,
- `Valor`$(i) =$ `BIG` para $i > n$.

Só que n é desconhecido. No fundo, `VObsc` é um vetor ordenado com vergonha do tamanho.
Descreva um algoritmo que, dado um inteiro k, e um objeto da classe `VObsc`, devolve um inteiro i tal que `Valor`$(i) = k$ ou -1 se não existir tal i. Seu algoritmo deve consumir tempo $O(\log n)$; lembre-se que você não conhece o n, mas ele existe mesmo assim.

Questão 5 [2 pontos]
O problema SUBGRAFO tem como entrada dois grafos G e H, e pergunta se G tem um subgrafo isomorfo a H.

(a) Mostre que SUBGRAFO é NP-completo, com uma redução a partir de 3–SAT, CIRCUITO HAMILTONIANO ou CONJUNTO INDEPENDENTE MÁXIMO, à sua escolha.

(b) Dado um grafo H, o problema SUBGRAFO–H tem como entrada um grafo G e pergunta se G tem um subgrafo isomorfo a H. Mostre que este problema é polinomial ou que é NP-completo. Poderia ser ambos?

Questão 6 [2 pontos]
Imagine que você deve escolher entre os seguintes três algoritmos:

(a) Algoritmo A resolve um problema de tamanho n derivando dele cinco subproblemas de tamanho $n/2$, resolvendo-os recursivamente, e então combinando suas soluções em tempo linear para obter uma solução do problema original.

(b) Algoritmo B resolve um problema de tamanho n resolvendo recursivamente dois problemas de tamanho $n-1$ e então combinando suas soluções em tempo constante para obter uma solução do problema original.

(c) Algoritmo C resolve um problema de tamanho n derivando dele nove subproblemas de tamanho $n/3$, resolvendo-os recursivamente e então combinando suas soluções em tempo $\mathrm{O}(n^2)$ para obter uma solução do problema original.

Qual é o consumo de tempo de cada um destes algoritmos (em notação assintótica)? Qual deles você escolheria?

Questão 7 [3 pontos]
Dada uma quantidade ilimitada de moedas nos valores $x_1, x_2, \ldots, x_n$, desejamos dar, nestas moedas, um troco no valor de v. Ou seja, desejamos encontrar um (multi)conjunto de tais moedas cujo valor total é v. Talvez isso não seja possível: por exemplo, se as moedas são de valores 5 e 10 apenas, então podemos fazer um troco de 15, porém não de 12. Escreva um algoritmo de programação dinâmica que consuma tempo $\mathrm{O}(nv)$ para resolver o seguinte problema:

> **Entrada:** n, $x_1, x_2, \ldots, x_n$ e v.
> **Questão:** é possível troco de v com moedas de valores $x_1, x_2, \ldots, x_n$?

Esse algoritmo é polinomial no tamanho da entrada? Elabore sobre isso.

Questão 8 [3 pontos]
Dada uma matriz $m \times n$ de bits A, e um vetor de m bits b, não existe, em geral, um vetor de n bits x tal que $Ax = b$, onde os bits são entendidos como elementos do corpo dos inteiros módulo 2. Entretanto, mostra-se que um vetor aleatório satisfaz, em média, pelo menos $\lceil m/2 \rceil$ das equações dadas pelas linhas do sistema $Ax = b$. Dê um algoritmo determinístico que encontre um x com essa propriedade, em *tempo linear* no tamanho de A. Demonstre correção e complexidade.