MAC 5811
Projeto e Análise de Algoritmos

DCC–IME–USP, 7 de agosto de 2008
das 13 às 18 horas

1. Esta prova contém 8 questões, sendo 3 de um ponto, 4 de dois pontos e 1 de três pontos.
2. Você pode resolver todas as questões. A banca considerará questões cujos valores somem até 10 pontos de tal modo que sua nota seja a maior possível. Você precisa obter nota pelo menos 7 para ser aprovado.
3. Todas as suas afirmações devem ser *precisas* e devidamente *justificadas*.
4. Ao apresentar um algoritmo, você deve explicar o seu funcionamento em palavras e, se necessário, descrevê-lo em pseudocódigo.
5. Você pode utilizar como subrotina qualquer algoritmo do CLRS[a] sem reescrevê-lo, mesmo que o algoritmo necessite de alguma pequena alteração. No entanto, você deve descrever a eventual alteração e dizer claramente o que o algoritmo recebe, o que devolve e o que faz. Deve dizer também qual o consumo de tempo do algoritmo.
6. Não é permitida a consulta a livros, anotações, colegas, calculadoras, Internet, computadores etc.
7. Comece a responder cada questão em uma folha nova. Não escreva no verso das folhas. Numere as folhas.

[a] Livro de Cormen, Leiserson, Rivest e Stein.

Questão 0 [0 pontos]
Leia as instruções.

Questão 1 [1 ponto]
É verdade que $2^{n+1} = \Theta(2^n)$? É verdade que $3^n = \Theta(2^n)$? É verdade que $2^{2n} = \Theta(2^n)$?

Questão 2 [1 ponto]

Suponha que estamos estudando o desempenho de um algoritmo em função do tamanho, n, das instâncias de um problema. Considere as seguintes afirmações:

(1) "o consumo de tempo do algoritmo é $O(n^2)$ no pior caso",

(2) "o consumo de tempo do algoritmo é $O(n^2)$ para toda instância do problema",

(3) "o consumo de tempo do algoritmo é $O(n^2)$ no melhor caso" e

(4) "o consumo de tempo do algoritmo é $O(n^2)$ para alguma instância de tamanho n".

Qual a diferença entre as afirmações (1) e (2)? Qual a diferença entre as afirmações (3) e (4)?

Questão 3 [2 pontos]

Seja T a função definida pela seguinte recorrência: $T(1) = 1$ e

$$T(n) = T(\lfloor\sqrt{n}\rfloor) + 1$$

para todo inteiro $n > 1$. Mostre que $T(n) = \Theta(\lg\lg n)$. (Sugestão: Comece por restringir a recorrência aos números da forma 2^{2^i}.)

Questão 4 [2 pontos]

Dado um vetor $S[1\,..\,n]$ de números inteiros distintos dois a dois, queremos construir um vetor crescente $B[1\,..\,i]$ contendo os i menores elementos de S. Para cada uma das idéias abaixo, dê um algoritmo com o melhor desempenho assintótico de pior caso que você puder. Analise o consumo de tempo de cada algoritmo em função de n e i.

a. Recolha os i primeiros elementos de S depois de arranjar S em ordem crescente.

b. Construa um heap com os elementos de S e chame a função ExtractMin i vezes.

c. Encontre o valor do i-ésimo menor elemento de S, faça uma partição de S em torno desse valor, e finalmente rearranje os i menores elementos em ordem crescente.

Questão 5 [3 pontos]

Temos n estações ao longo de um rio, numeradas de 1 a n na direção da correnteza. Você pode alugar um barco em qualquer estação i, descer o rio até uma estação $k > i$, devolver o barco e pagar uma taxa $c(i, k)$ pelo passeio. É possível que $c(i, k)$ seja maior que $c(i, j) + c(j, k)$, sendo j uma estação intermediária entre i e k. Nesse caso, é mais barato alugar um barco de i a j e depois outro de j até k.

Dê um algoritmo eficiente que calcule o custo mínimo de uma viagem de 1 a n. Exiba a recorrência que serve de base para o seu algoritmo. Em função de n, quanto tempo o seu algoritmo consome?

Questão 6 [2 pontos]
Digamos que um vetor $A[1\,..\,n]$ está *arrumado* se existe um índice i em $1\,..\,n$ tal que

$$A[1\,..\,i-1] \leq A[i] \leq A[i+1\,..\,n]$$

(Expressões da forma "$A[h\,..\,k] \leq x$" significam "$A[j] \leq x$ para $j = h, \ldots, k$".)

Escreva um algoritmo que decida se um vetor $A[1\,..\,n]$ está ou não arrumado. Em caso afirmativo, o seu algoritmo deve devolver um índice i que satisfaz as condições da definição. O seu algoritmo deve consumir tempo $O(n)$.

Questão 7 [1 ponto]
Uma *subseqüência* de uma seqüência $\langle a_1, \ldots, a_n \rangle$ é o que sobra depois que um conjunto arbitrário de termos de $\langle a_1, \ldots, a_n \rangle$ é apagado.

Escreva um algoritmo linear para resolver o seguinte problema: dadas duas seqüências Z e X, decidir se Z é subseqüência de X. Prove que o seu algoritmo está correto.

Questão 8 [2 pontos]
Problema 1: Dados números inteiros positivos $w_1, \ldots, w_n$, decidir se existe uma partição (A, B) de $\{1, \ldots, n\}$ tal que $\sum_{i \in A} w_i = \sum_{i \in B} w_i$.

Problema 2: Dado um número s e números inteiros positivos $w_1, \ldots, w_n$, decidir se existe uma parte A de $\{1, \ldots, n\}$ tal que $\sum_{i \in A} w_i = s$.

Sabe-se que o Problema 2 é NP-completo. Mostre que o Problema 1 também é NP-completo.