MAC5811 Projeto e Análise de Algoritmos

DCC–IME–USP, 1 de agosto de 2007

Instruções

(i) Esta prova contém oito questões sendo todas de dois pontos.

(ii) A banca considerará questões cujos valores somem até 10 pontos de modo que a soma total das notas obtidas seja máxima. Um aluno, para ser aprovado, precisa obter nessas questões pelo menos 7 pontos.

(iii) Enuncie os teoremas e propriedades usados para justificar suas afirmações.

(iv) Não é permitida a consulta a livros, anotações, colegas, calculadoras, internet, computadores ...

Duração da prova: 5 horas

Questão 1 [2 pontos] CLRS 4.3-2
Um algoritmo ALG-A tem o seu consumo de tempo descrito através da recorrência

$$T(n) = 7\,T(n/2) + n^2.$$

Um outro algoritmo ALG-B tem o seu consumo de tempo descrito através da recorrência

$$T'(n) = a\,T'(n/4) + n^2,$$

onde a é uma constante. Qual é o maior valor inteiro possível para a de tal forma que o algoritmo ALG-B seja assintoticamente mais eficiente que o algoritmo ALG-A. Justifique a sua resposta.

Questão 2 [2 pontos] CLRS 2-4
Dado um vetor $A[1\,..\,n]$ de inteiros, chamamos de uma **inversão** de A um par de índices (i, j) do vetor A tal que $i < j$ e $A[i] > A[j]$.

Escreva um algoritmo CONTA-INVERSÕES(A, n) que recebe um vetor $A[1\,..\,n]$ com inteiros todos distintos e devolve o número de inversões de A. Seu algoritmo deve consumir tempo $O(n \lg n)$. Explique sucintamente porque seu algoritmo está correto e tem o consumo de tempo pedido.

Questão 3 [2 pontos] CLRS 9.3-5
Suponha que MEDIANA(A, p, r) é um algoritmo que rearranja os elementos de um dado vetor $A[p\,..\,r]$ de números inteiros e devolve um índice q, onde $p \leq q \leq r$, tal que

$$A[p\,..\,q-1] \leq A[q] \leq A[q+1\,..\,r]$$

e $A[q]$ é uma mediana de $A[p\,..\,r]$. Suponha ainda que o consumo de tempo do algoritmo MEDIANA é linear.

Escreva um algoritmo SELECT(A, p, r, i) que recebe um vetor $A[p\,..\,r]$ de números inteiros e um número inteiro i, onde $1 \leq i \leq r - p + 1$, e devolve o valor do i-ésimo menor elemento de $A[p\,..\,r]$. O seu algoritmo deve utilizar o algoritmo MEDIANA como subrotina e deve consumir tempo linear. Explique sucintamente porque seu algoritmo está correto e tem o consumo de tempo pedido.

Questão 4 [2 pontos] CLRS 5.1-2
Considere o algoritmo abaixo, que recebe dois números inteiros a e b tais que $0 \leq a \leq b$ e devolve um número inteiro i, onde $a \leq i \leq b$. O algoritmo usa a subrotina BIT-ALEATÓRIO() que devolve 1 ou 0, independentemente dos valores previamente devolvidos, com probabilidade uniforme.

RANDOM(a, b)
$n \leftarrow b - a$
$t \leftarrow \lceil \lg(n+1) \rceil$
repita
    $r \leftarrow 0$
    para $j \leftarrow 1$ **até** t **faça**
        $r \leftarrow 2 \times r +$ BIT-ALEATÓRIO()
até que $r \leq n$
$i \leftarrow a + r$
devolva i

Supondo que o consumo de tempo da subrotina BIT-ALEATÓRIO() é constante, qual o consumo de tempo esperado do algoritmo RANDOM? Use a notação O, mas procure dar a resposta o mais justa possível e justifique-a.

Questão 5 [2 pontos]
Uma vez, em um império antigo, havia duas torres de formatos desiguais em duas cidades diferentes. As torres foram construídas colocando-se lajotas circulares umas sobre as outras. As lajotas tinham todas a mesma altura, e o raio de cada uma delas era um número inteiro. Não era de se espantar que, apesar das torres terem formatos diferentes, elas possuiam várias lajotas de mesmo raio. Mais de mil anos após a sua construção, o imperador ordenou que seus arquitetos removessem algumas das lajotas de cada uma das torres de modo que elas ficassem exatamente com o mesmo formato, e ao mesmo tempo permanecessem tão altas quanto possível. A ordem das lajotas nas novas torres devia permanecer a mesma que nas torres originais. O imperador achava que, desta maneira, as duas torres se tornariam o símbolo da harmonia entre as duas cidades. Ele decidiu ainda que elas passariam a se chamar as **Torres Gêmeas**.

Agora, mais de dois mil anos depois, você foi desafiado a escrever um algoritmo

TORRES-GEMÊAS(A, n, B, m)

que recebe dois vetores $A[1 \,..\, n]$ e $B[1 \,..\, m]$ de inteiros, com o raio de cada uma das lajotas de duas torres na ordem em que elas aparecem nas torres, e devolve a altura (i.e., o número de lajotas) das maiores torres gêmeas que podem ser construídas a partir destas duas torres.

Seu algoritmo deve ser o mais eficiente possível. Explique sucintamente porque seu algoritmo está correto e diga qual é o consumo de tempo dele.

Questão 6 [2 pontos]
O seguinte algoritmo recebe o início de uma lista ligada e devolve VERDADEIRO se a lista ligada possui um ciclo e FALSO caso contrário.

TEM-CICLO(p)
1 $p_1 \leftarrow p$ $\quad$ $p_2 \leftarrow p$
2 **enquanto** $p_2 \neq$ NIL **faça**
3 $\quad$ $p_2 \leftarrow prox(p_2)$
4 $\quad$ **se** $p_2 = p_1$
5 $\quad\quad$ **então devolva** VERDADEIRO
6 $\quad$ **se** $p_2 \neq$ NIL
7 $\quad\quad$ **então** $p_2 \leftarrow prox(p_2)$
8 $\quad$ **se** $p_2 = p_1$
9 $\quad\quad$ **então devolva** VERDADEIRO
10 $\quad$ $p_1 \leftarrow prox(p_1)$
11 **devolva** FALSO

(a) Explique porque o algoritmo está correto.

(b) Denote por n o número de elementos na lista apontada por p. Quantas vezes a linha 2 é executada no pior caso? Não use notação assintótica. Dê a resposta mais precisa que você conseguir. Justifique cuidadosamente a sua resposta.

Questão 7 [2 pontos]
Considere o seguinte algoritmo que recebe um vetor $E[1 \mathinner{..} n]$ e devolve um vetor de números inteiros $S[1 \mathinner{..} n]$.

FAZ-ALGO(E, n)
$j \leftarrow 1$
$t \leftarrow 0$
$S[1] \leftarrow 0$
enquanto $j < n$ **faça**
$\quad$ **enquanto** $t > 0$ **e** $E[t] \neq E[j]$ **faça**
$\quad\quad$ $t \leftarrow S[t]$
$\quad$ $t \leftarrow t + 1$
$\quad$ $j \leftarrow j + 1$
$\quad$ **se** $E[j] = E[t]$
$\quad\quad$ **então** $S[j] \leftarrow S[t]$
$\quad\quad$ **senão** $S[j] \leftarrow t$
devolva S

Um aluno alega que o consumo de tempo de tal algoritmo é $O(n)$, independentemente do conteúdo do vetor $E[1 \mathinner{..} n]$. O aluno está correto? Justifique a sua resposta.

Questão 8 [2 pontos]
Considere o problema a seguir.

> **Problema** PARTIÇÃO(A, n): Dado um vetor $A[1\,..\,n]$ de números inteiros positivos, decidir se existe um subconjunto I de $\{1, \ldots, n\}$ tal que
>
> $$\sum_{i \in I} A[i] = \sum_{i \notin I} A[i] \ .$$

O seguinte algoritmo resolve o problema PARTIÇÃO.

```
TEM-PARTIÇÃO(A, n)
s ← 0
para i ← 1 até n faça s ← s + A[i]
se s é ímpar então devolva FALSO
T[0, 0] ← VERDADEIRO
para i ← 1 até n faça T[0, j] ← FALSO
para i ← 1 até n faça
    T[i, 0] ← VERDADEIRO
    para j ← 1 até s faça
        T[i, j] ← T[i − 1, j]
        se A[i] > j e T[i − 1, j − A[i]] = VERDADEIRO
            então T[i, j] ← VERDADEIRO
devolva T[n, s/2]
```

(a) Qual é o consumo de tempo do algoritmo TEM-PARTIÇÃO?

(b) Sabe-se que o problema PARTIÇÃO é NP-completo. Isso contradiz a análise do item anterior? Justifique cuidadosamente a sua resposta.