Exame de MAC5811 - Projeto e Análise de Algoritmos

DCC–IME–USP, julho, 2004

Instruções:

(i) O candidato pode resolver todas as questões.

(ii) Esta prova contém três questões de um ponto, três de dois pontos e duas de três pontos.

(iii) 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.

(iv) Mencione os teoremas e propriedades usados para justificar suas afirmações.

Duração do exame: 5 horas

Questão 1 [1 ponto]
Diga se são verdadeiras ou falsas as seguintes afirmações e justifique sucintamente as suas respostas. Sejam f e g funções de $\mathbf{N}$ em $\mathbf{N}$.

a) $f(n) = O(g(n))$ implica que $2^{f(n)} = O(2^{g(n)})$.

b) $f(n) = O(n)$ e $g(n) = O(f(n))$ implica que $g(n) = O(n)$.

c) $f(n) = \Omega(n)$ e $g(n) = \mathrm{O}(f(n))$ implica que $g(n) = \mathrm{O}(n)$.

d) $f(n) = \Omega(n)$ e $g(n) = \mathrm{O}(f(n))$ implica que $g(n) = \Omega(n)$.

Questão 2 [1 ponto]
Meu manual diz que um problema "está em NP"; concluo que não existe algoritmo polinomial para o problema. Estou certo ou errado? Justifique.

Questão 3 [1 ponto]
Considere o seguinte algoritmo que recebe um número natural n e devolve 0 se n é primo e um fator não-trivial de n se n não é primo.

Fator(n)
para $d \leftarrow 2$ **até** $n-1$ **faça**
    se resto$(n/d) = 0$
        devolva d
devolva 0

Um aluno alega que o consumo de tempo deste algoritmo não é polinomial. Ele está certo? Justifique a sua resposta.

Questão 4 [2 pontos]
Considere um vetor $V[1\,..\,n]$ de inteiros tal que, para todo i, vale $|V[i] - V[i+1]| \leq 1$. Escreva um algoritmo $\textsc{Busca}(V, n, z)$ que, dado um vetor $V[1\,..\,n]$ como acima e um inteiro z tais que $V[1] \leq z \leq V[n]$, determina um índice j em $\{1, \ldots, n\}$ tal que $V[j] = z$. Seu algoritmo deve consumir tempo $\mathrm{O}(\lg n)$. Explique sucintamente porque seu algoritmo está correto e tem a complexidade de tempo pedida.

Questão 5 [2 pontos]
Considere a seguinte variante do algoritmo $\textsc{Quicksort}$, que recebe e ordena um vetor $A[p\,..\,r]$. O $\textsc{Particione}$ usa o elemento $x := A[r]$ como pivô, rearranja os elementos de $A[p\,..\,r]$ e devolve um índice q em $\{p, \ldots, r\}$ tal que x fica na posição q, os elementos menores ou iguais a x ficam nas posições de p a $q-1$ e os elementos maiores que x ficam nas posições de $q+1$ a r.

$\textsc{Quicksort2}(A, p, r)$
enquanto $p < r$ **faça**
    $q \leftarrow \textsc{Particione}(A, p, r)$
    $\textsc{Quicksort2}(A, p, q-1)$
    $p \leftarrow q + 1$

Mostre que a pilha de recursão pode atingir altura proporcional a n, onde $n := r - p + 1$. Modifique o código de modo que a pilha de recursão tenha altura $\mathrm{O}(\lg n)$. Justifique a sua resposta.

Questão 6 [2 pontos]
Uma seqüência $X[1\,..\,m]$ é subseqüência de uma seqüência $Y[1\,..\,n]$ se existem índices $i_1 < i_2 < \ldots < i_k$ tais que

$$X[j] = Y[i_j] \quad \text{para } j = 1, \ldots, k.$$

Por exemplo, $X = \mathtt{BCDB}$ é uma subseqüência de $Y = \mathtt{A\underline{B}\underline{C}D\underline{D}A\underline{B}}$ com índices $2 < 3 < 5 < 7$.

Escreva uma função $\textsc{Subseq-Cont}(X, m, Y, n)$ que recebe uma seqüência $X[1\,..\,m]$ e uma seqüência $Y[1\,..\,n]$, e devolve o número de ocorrências de X como subseqüência de Y. Sua função deve consumir tempo $\mathrm{O}(mn)$.

Exemplos:

a) Se $X = \mathtt{RABBIT}$ e $Y = \mathtt{RABBBIT}$, sua função deve devolver 3.

b) Se $X = \mathtt{BAG}$ e $Y = \mathtt{BABGBAG}$, sua função deve devolver 5.

Argumente porque sua função produz a resposta correta e mostre que o consumo de tempo é de fato $\mathrm{O}(mn)$.

Sugestão: Considere a quantidade

$$c[i, j] = \text{ número de ocorrências de } X[1\,..\,i] \text{ em } Y[1\,..\,j].$$

Expresse $c[i, j]$ através de uma recorrência.

Questão 7 [3 pontos]
Considere o seguinte algoritmo, cujo argumento n é uma potência de 2. O algoritmo não faz nada de útil.

ALGO(n)
se $n = 1$ **então devolva** 1
para $i \leftarrow 1$ **até** 8 **faça** $z \leftarrow$ ALGO($n/2$)
para $i \leftarrow 1$ **até** n^3 **faça** $z \leftarrow 0$

a) Seja $T(n)$ o número de vezes que a atribuição "$z \leftarrow 0$" é executada. Escreva uma recorrência que defina $T(n)$.

b) Mostre diretamente (sem usar o Master Theorem) que $T(n)$ é $\Omega(n^3 \lg n)$.

c) Troque "8" por "7" no algoritmo e mostre diretamente que $T(n)$ é $\mathrm{O}(n^3)$.

Questão 8 [3 pontos]
Considere o seguinte algoritmo:

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

Um aluno alega que o consumo de tempo de tal algoritmo é $O(n)$. O aluno está correto? Justifique a sua resposta.

BOA PROVA!