Rio de Janeiro, 13 de Setembro de 2010

PROVA 1 DE ANÁLISE DE ALGORITMOS

PROFESSOR: EDUARDO SANY LABER

DURAÇÃO: 1:50h

1. (2.0pt) Considere o problema $\mathcal{P}$ com a seguinte especificação.

Entrada: vetor A contendo n inteiros pertencentes ao conjunto $\{1, ..., k\}$, aonde $k \leq n$.

Saida: vetor B de k posições, aonde $B[i]$, $1 \leq i \leq k$, deve armazenar a soma de todos os números do vetor A que são menores ou iguais a i.

a) Escreva o pseudo-código de um algortimo para resolver o problema $\mathcal{P}$ e analise a sua complexidade assintótica como função de n. Algoritmos eficientes receberão mais pontos.

b) Assuma agora que temos disponível o vetor B corretamente preenchido. Dados dois inteiros a e b, com $a \leq b$, com que complexidade podemos determinar a soma de todos os inteiros do vetor A que pertencem ao intervalo $[a, b]$? Por que?

2. (2.0pt) Seja $S = \{a_1, \ldots, a_n\}$ um conjunto de n números distintos e um inteiro x. Considere o problema $\mathcal{P}$ de determinar se existem três números distintos em S cuja soma é x.

a) Seja $T(n)$ a complexidade de pior caso do algoritmo Alg, apresentado abaixo, para resolver $\mathcal{P}$. Encontre $f(n)$ tal que $T(n) = \Theta(f(n))$.

Proc Alg
    Para i=1,...,n-2
        Para j=i+1,...,n-1
            Para k=j+1,...,n
                Se a_i+a_j+a_k =x
                    Return SIM
    Return NÃO
Fim

b) Projete um algoritmo com complexidade $O(n^2 \log n)$ para resolver o problema $\mathcal{P}$. Não é necessário apresentar o pseudo-código mas sim explicar com clareza os passos que o algoritmo deve realizar e explicar a complexidade.

3. (3.0pt) Considere o seguinte procedimento para manipular uma lista A.

Proc(A: lista)
    Se $|A|$=1
        Return
    Senão
        Ordene(A)
        Particione A em duas listas A_e e A_d, cada uma delas com tamanho $|A|/2$.
        Proc(A_e)
        Proc(A_d)
    Fim Se
Fim Proc

a) Seja $T(n)$ o número de operações realizadas por Proc para a pior instância de tamanho n. Assuma que a partição em A_e e A_d é arbitrária e que esta consome $O(n)$ operações. Além disso, assuma que $ordene(A)$ execute $g(n)$ operações para pior instância de tamanho n. Escreva uma equação de recorrência para $T(n)$.

b) Resolva a recorrência para o caso em que Ordene é o BubbleSort.

c) Resolva a recorrência para o caso em que Ordene é o HeapSort.

4. (2.0pt) Seja A um algoritmo que resolve um problema $\mathcal{P}$ e seja $T(n)$ a função que define a complexidade de pior caso de A em função do tamanho da entrada n.

a) Assuma $T(n) \in O(n^3)$. Podemos afirmar que existe $n_0 > 1$ tal que para toda entrada de tamanho n, com $n \geq n_0$, o algoritmo gasta no máximo n^3 operações? Por que?

b) Assuma $T(n) \in \Theta(n^3)$. Podemos afirmar que existe $n_0 > 1$ tal que para alguma entrada de tamanho n, com $n \geq n_0$, o algoritmo gasta pelo menos $1000n^2$ operações? Por que?

5. (3.0pt) Considere o seguinte problema P

Entrada: três vetores ordenados A, B, e C, contendo números inteiros.

Saida: SIM se existe uma tripla de inteiros (i, j, k) tal que a sequência $(A[i], B[j], C[k])$ forma uma progressão aritmética de razão 5; NÃO, caso contrário.

Seja n a soma dos tamanhos dos três vetores, $n = |A| + |B| + |C|$. Descreva um algoritmo eficiente para resolver $\mathcal{P}$ e analise a sua complexidade assintótica em função de n. Não é necessário apresentar o pseudo-código mas sim explicar com clareza os passos que o algoritmo deve realizar. Novamente, o número de pontos será proporcional a eficiência do algoritmo,