MAC5811 Projeto e Análise de Algoritmos

DCC–IME–USP, 26 de fevereiro de 2008

Instruções

(*i*) Esta prova contém nove questões, sendo seis de dois pontos e três de três 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*) Todas as suas afirmações devem ser **precisas** e **justificadas**.

(*iv*) Para descrever um algoritmo você deve escrever em palavras como ele funciona e, se necessário, escrever um pseudo-código.

(*v*) Você pode utilizar como subrotina qualquer algoritmo visto em sala de aula sem reescrevê-lo, mesmo que o algoritmo necessite de alguma **pequena** alteração. No entanto, você deve descrever clara e sucintamente o que o algoritmo recebe, devolve ou faz e o seu consumo de tempo e qual é essa alteração. Exemplo

> *"O algoritmo* BLÁ-BLÁ-BLÁ *usa como subrotina o algoritmo* ORDENAÇÃO-LERDA (A, n) *que recebe e rearranja um vetor* $A[1\,..\,n]$ *de modo que ele fique em ordem crescente. O consumo de tempo do algoritmo* ORDENAÇÃO-LERDA *é* $\mathrm{O}(n^n)$*."*

(*vi*) Não é permitida a consulta a livros, anotações, colegas, calculadoras, Internet, computadores ...

Questão 1 [2 pontos] CLRS 3.1-3

1. Sejam $f(n)$ e $g(n)$ funções dos inteiros nos inteiros positivos. O que significa dizer que $f(n) = O(g(n))$?
2. Explique por que a afirmação "O consumo de tempo do algoritmo A é pelo menos $O(n^2)$" não faz sentido.

Questão 2 [2 pontos] CLRS 2.3-7

Descreva um algoritmo com consumo de tempo $O(n \lg n)$ que, dados um conjunto S de n inteiros e outro inteiro x, determina se existem ou não dois elementos em S cuja soma é exatamente x.

Questão 3 [2 pontos]

Um vetor $A[1 \dots n]$ é dito ter um *elemento majoritário* se mais do que a metade de suas entradas tem o mesmo valor. Portanto, x é elemento majoritário de A se o número de ocorrências de x em A é maior do que $n/2$. A tarefa é projetar um algoritmo que, dado um vetor com n entradas, determina se há um elemento majoritário e, se for o caso, encontra-o. O consumo de tempo de seu algoritmo deve ser $O(n \lg n)$. Os elementos do vetor não são necessariamente de algum conjunto ordenado, como os inteiros, e não pode haver comparação do tipo "$A[i] > A[j]$?". Entretanto, questões do tipo "$A[i] = A[j]$?" podem ser respondidas em tempo constante.

Questão 4 [2 pontos]

Para cada um dos algoritmos abaixo, escreva a recorrência para o consumo de tempo no pior caso e a resolva usando notação assintótica.

1. Algoritmo A_1 tem como entrada uma matriz $n \times n$. O algoritmo executa 8 chamadas recursivas para matrizes de tamanho $n/2 \times n/2$. O consumo de tempo do restante do processamento é dominado por uma chamada do algoritmo de Strassen para multiplicar duas matrizes $3n \times 3n$. Essencialmente, a cada 8 chamadas recursivas é feita uma chamada ao algoritmo de Strassen. Você pode supor que n é potência de 2.
2. Algoritmo A_2 tem como entrada uma seqüência de n inteiros. Ele faz 3 chamadas recursivas, cada uma delas para seqüência de $n/3$ inteiros. O consumo de tempo do restante do processamento é dominado por uma chamada do algoritmo mergesort para ordenar uma seqüência de n inteiros. Essencialmente, a cada 3 chamadas recursivas é feita uma chamada ao algoritmo mergesort. Você pode supor que n é potência de 3.

Questão 5 [2 pontos] CLRS 9-2

Escreva um algoritmo $\text{SELECT}(A, W, p, r, i)$ que recebe vetores $A[p \,..\, r]$ e $W[p \,..\, r]$ de números inteiros positivos e um número inteiro i tal que $1 \le i \le \sum_{j=p}^{r} W[j]$ e devolve o valor do i-ésimo menor elemento da lista formada por $W[p]$ cópias de $A[p]$, $W[p+1]$ cópias de $A[p+1]$, ..., $W[r]$ cópias de $A[r]$. Por exemplo, para $p = 6, r = 8, i = 6$,

5	6	7	8	9	
...	4	2	8	...	A ,

e

5	6	7	8	9	
...	3	5	4	...	W

o algoritmo deve devolver 4 que é o 6º menor elemento da lista formada por

$$4 \;\; 4 \;\; 4 \;\; 2 \;\; 2 \;\; 2 \;\; 2 \;\; 2 \;\; 8 \;\; 8 \;\; 8 \;\; 8 \,.$$

O consumo de tempo do algoritmo SELECT deve ser $O(n)$, onde $n = r - p + 1$. Explique sucintamente por que seu algoritmo está correto e tem o consumo de tempo pedido.

Questão 6 [2 pontos] CLRS 16-4

Seja $\{1, \ldots, n\}$ um conjunto de **tarefas**. Cada tarefa consome um dia de trabalho; durante um dia de trabalho somente uma das tarefas pode ser executada. Os dias de trabalho são numerados de 1 a n. A cada tarefa t está associado um **prazo** $p[t]$: idealmente, a tarefa t deve ser executada em algum dia do intervalo $1 \mathinner{..} p[t]$. A cada tarefa t está associada uma **multa** não-negativa $m[t]$. Se uma dada tarefa t é executada depois do prazo $p[t]$, a multa $m[t]$ deverá ser paga (mas a multa não depende do número de dias de atraso). Um **escalonamento** de tarefas é um vetor e tal que $e[d] = t$ se a tarefa t deve ser executada no dia d.

Escreva um algoritmo MIN-MULTA(p, m, n) que recebe um vetor de prazos $p[1 \mathinner{..} n]$ e um vetor de multas $m[1 \mathinner{..} n]$ e devolve o menor valor da multa de um escalonamento. O consumo de tempo do seu algoritmo deve ser $O(n^2)$.

Questão 7 [3 pontos] CLRS 34-2

Bonnie e Clyde acabaram de roubar um banco. Eles têm um saco de dinheiro que precisa ser dividido. Para cada uma das possibilidades abaixo, descreva um algoritmo polinomial ou mostre que o problema é NP-completo. A entrada é sempre uma lista de n itens, junto com o valor de cada item. Use que o problema da *Partição* é NP-difícil: *dado um conjunto de inteiros positivos X, decidir se existe um subconjunto de X cuja soma é a metade da soma dos elementos de X.*

1. O saco contém n moedas, mas somente 2 possíveis valores: algumas valem x reais e outras valem y reais. Eles desejam dividir o dinheiro em partes iguais.
2. O saco contém n cheques que, por uma coincidência espantosa, estão nominais a "Bonnie ou Clyde". Eles desejam dividir o dinheiro em partes iguais.
3. O saco contém n cheques, como no item anterior, mas desta vez eles aceitam uma diferença de 100 reais nos totais que cada um recebe.

Questão 8 [3 pontos] CLRS 8-5

Suponha que, em vez de ordenar um vetor, precisamos apenas que os elementos cresçam em média. Mais precisamente, um vetor $A[1 \ldots n]$ com n elementos é dito estar *k-ordenado* se, para cada $i = 1, 2, \ldots, n - k$, o seguinte vale:

$$\frac{1}{k} \sum_{j=i}^{i+k-1} A[j] \leq \frac{1}{k} \sum_{j=i+1}^{i+k} A[j].$$

a. O que quer dizer que um vetor é 1-ordenado?

b. Mostre uma permutação dos números $1, 2, \ldots, 10$ que está 2-ordenado, mas não ordenado.

c. Mostre que A está k-ordenado se e somente se $A[i] \leq A[i + k]$ para cada $i = 1, 2, \ldots, n - k$.

d. Descreva um algoritmo com consumo de tempo $O(n \lg(n/k))$ que k-ordene um vetor com n elementos.

e. Mostre que um vetor k-ordenado pode ser ordenado em tempo $O(n \lg k)$.

f. Mostre que quando k é constante, o consumo de tempo para k-ordenar um vetor é $\Omega(n \lg n)$ no pior caso.

Questão 9 [3 pontos] CLRS 22.4-2
Seja $G = (V, A)$ um grafo orientado acíclico com pesos positivos nas arestas dados por $w : A \longrightarrow \mathbb{R}$.
O problema da *contagem de caminhos* é o seguinte: *dados dois vértices u e v em V, encontrar o número de caminhos que conectam u a v.*
O problema da *contagem de caminhos mínimos* é o seguinte: *dados dois vértices u e v em V, encontrar o número de caminhos de comprimento mínimo que conectam u a v.*
Por exemplo, no grafo abaixo existem 6 caminhos de u a v, dentre os quais somente 2 são caminhos mínimos.

a. Descreva um algoritmo eficiente para o problema da *contagem de caminhos*. Analise a sua complexidade.

b. Descreva um algoritmo eficiente para o problema da *contagem de caminhos mínimos*. Analise a sua complexidade.