Departamento de Ciência de Computadores FCUP

Algoritmos (CC4010) 2015/16

Teste (07.01.2016) *duração: 3h*

N.º [ ] Nome [ ]

1. Seja $T(n)$ uma função não negativa e crescente tal que $T(n) = T(a_1 n) + T(a_2 n) + \cdots T(a_k n) + bn$, onde k é um inteiro positivo fixo, $b \geq 1$ e $a_1 + a_2 + \cdots + a_k < 1$, sendo $a_1, a_2, \ldots a_k, b$ constantes reais positivas. Assuma que $T(n) \in O(1)$ para valores suficientemente pequenos de n.

a) Prove que $T(n) \in \Omega(n)$, diretamente a partir da definição formal de $\Omega(n)$.

b) Por análise da árvore de recursão, prove que $T(n) \in \Theta(n)$.

2. Considere uma *stack* suportada por um *array* V e uma variável *top* que designa o índice da primeira posição livre de V, com as operações usuais PUSH(x) e POP() assim definidas:

```
PUSH(x)                    POP()
| V[top] := x              | top := top − 1
| top := top + 1           | retorna V[top]
```

Supondo que inicialmente V tem M posições, considere a possibilidade de a operação de PUSH(x) ser modificada para permitir aumentar a capacidade da stack de M posições se esta se encontrar cheia (realocando espaço e copiando os elementos que estavam na stack para o novo array).

Considere um modelo de custos em que a inserção de um elemento e a remoção de um elemento têm custo 1 e o custo da expansão da estrutura de dados é igual ao número de elementos que é necessário transferir.

a) Suponha que $M = 20$ e que efetua uma sequência de 102 operações de PUSH. Apresente a expressão que define o custo total. Qual é o custo de uma operação de PUSH no pior caso e no melhor caso? Qual é o custo amortizado?

b) Considere o caso geral (em que apenas se sabe que M é uma constante). Suponha que efetua uma sequência de kM operações de PUSH. Qual é o custo amortizado de cada operação? Considerando esse valor, optaria por esta estratégia de expansão face a outras estudadas?

3. Supondo que P $\neq$ NP, classifique os problemas enunciados em cada alínea como membros ou não das classes de problemas P, NP e NP-completos, justificando (pode usar conhecimentos sobre outros problemas).

a) Dados n, uma sequência $c_1, \ldots, c_n$ de inteiros positivos, e um valor k, decidir se existe $I \subseteq \{1, \ldots, n\}$ tal que $|I| \geq k$ e $c_i \geq c_j$, qualquer que seja $i \in I$ e $j \in \{1, \ldots, n\}$, sendo $|I|$ o número de elementos de I.

b) Dado um grafo não dirigido $G = (V, E)$ e dada uma função $p : V \to \mathbb{Z}^+$ que associa um peso a cada vértice de G e ainda um valor $P \in \mathbb{Z}^+$, decidir se existe um conjunto de vértices $W \subseteq V$ tal que qualquer que seja a aresta $e \in E$, algum dos extremos de e pertence a W, e $\sum_{v \in W} p(v) \leq P$.

(Continua, v.p.f.)

4. Suponha que lhe é apresentado um algoritmo polinomial que, aplicado a qualquer instância I de um certo problema de optimização, produz sempre uma solução s_I que satisfaz as restrições dessa instância e que verifica $r\,C(s_I) \leq C_{OPT}(I)$, sendo $r > 1$, uma constante fixa. Comente a proposta, considerando que se trata de um problema de minimização do custo e que $C(s_I)$ e $C_{OPT}(I)$ designam o custo de s_I e o custo óptimo das soluções da instância.

5. Dê exemplo de um problema da classe NPO que pertence à classe APX. Dê exemplo de um problema da classe NPO que não pertence à classe APX se $P \neq NP$.

6. Seja $\mathcal{P} = \{p_1, \ldots, p_n\}$ um conjunto de pontos do plano, sendo p_1 o ponto que tem menor ordenada. Admita que não existem três pontos colineares em P.

a) Suponha que $\mathcal{P}$ está ordenado por ordem crescente de ângulo polar relativamente a p_1. Baseando-se no algoritmo de Graham, escreva em pseudo-código um algoritmo **óptimo** para determinar os pontos de $\mathcal{P}$ que definem o seu invólucro convexo. Apresente a análise da complexidade do algoritmo.

b) Compare esse algoritmo com o algoritmo de Graham (*Graham scan*) para determinação do invólucro convexo de um qualquer conjunto $\mathcal{P}$ nas condições descritas no cabeçalho.

7. Suponha que estavam marcadas várias reuniões para um certo dia, sendo dada a hora de início e a duração de cada reunião. Admita que, por qualquer imprevisto, uma mesma pessoa tem de estar presente nessas reuniões (mas apenas numa única em cada momento). Havendo colisões, algumas reuniões terão de ser canceladas. Apresente uma estratégia greedy para selecionar um conjunto de reuniões a realizar de forma a minimizar o número de reuniões a cancelar.

8. Considere o problema "Minimum Edit Distance", de edição), em que os custos das operações de inserção e remoção são iguais a c mas o custo de uma substituição é $2c$, sendo $c > 0$ um inteiro (fixo).

a) Descreva sucintamente o problema.

b) Apresente a sua resolução por programação dinâmica, começando por indicar a recorrência em que se baseia.

9. Diga, justificando, qual é a complexidade de um algoritmo ótimo de seleção do $n/2$-ésimo menor elemento de um vetor de n elementos no caso de **a)** o vetor estar ordenado; **b)** que comece por ordenar o vetor usando um algoritmo baseado em comparação; **c)** que se baseie no algoritmo que utiliza a *mediana das medianas de cinco* descrito nas aulas.

(FIM)

Master theorem:

Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence $T(n) = aT(n/b) + f(n)$, where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

1. If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log_2 n)$.
3. If $f(n) = \Omega(n^{\log_b a + \varepsilon})$, for some constant $\varepsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.

Stirling's approximation:

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n (1 + \Theta(1/n)) = \sqrt{2\pi n}\left(\frac{n}{e}\right)^{\alpha_n}, \quad \text{with } 1/(12n+1) < \alpha_n < 1/(12n)$$

Some useful results:

$$\log(\prod_{k=1}^{n} a_k) = \sum_{k=1}^{n} \log a_k \qquad\qquad \sum_{k=0}^{\infty} kx^k = \frac{x}{(1-x)^2}, \text{ for } |x| < 1$$

If $(u_k)_k$ is an arithmetic progression (i.e., $u_{k+1} = r + u_k$, for some constant $r \neq 0$), then $\sum_{k=1}^{n} u_k = \frac{(u_1+u_n)n}{2}$.

If $(u_k)_k$ is a geometric progression (i.e., $u_{k+1} = ru_k$, for some constant $r \neq 1$), then $\sum_{k=1}^{n} u_k = \frac{u_{n+1}-u_1}{r-1}$.

If $f \geq 0$ is continuous and a monotonically increasing function, then

$$\int_{m-1}^{n} f(x)dx \leq \sum_{k=m}^{n} f(k) \leq \int_{m}^{n+1} f(x)dx$$