Departamento de Ciência de Computadores FCUP

Algoritmos (CC4010) 2015/16

Exame (19.01.2016) *duração: 3h*

N.º [ ] Nome [ ]

1. Usando **diretamente** a definição da classe $\Omega(8n^2 + n - 5)$, averigue se $2n^2 \in \Omega(8n^2 + n - 5)$.

2. Comente a afirmação: "qualquer algoritmo de ordenação que se baseie em comparação tem complexidade temporal $\Omega(n \log_2 n)$, no pior e no melhor caso, sendo n o número de inteiros a ordenar".

3. Prove que a complexidade amortizada de uma sequência de n operações de PUSH numa *stack* de inteiros, suportada por um *array*, com duplicação de espaço se necessário (por realocação), é $O(n)$.

4. Considere o problema "Coin Change" relativo à formação de uma quantia $q \geq 0$, usando um número mínimo de moedas de 1, 2, 5, 10, 20, 50, 100 e 200 cêntimos. Pretende-se *uma* solução ótima (quantas moedas utiliza no total e quantas moedas de cada tipo).

a) Admita que o número de moedas de cada tipo é *limitado*, sendo k_i o número das disponíveis do tipo t_i. Apresente um algoritmo para resolver o problema, usando *programação dinâmica (com memoização)*. Comece por indicar a **recorrência** subjacente (o que retorna se não existir solução?).

b) Admita que o número de moedas existentes de cada tipo é *ilimitado* (ou, se preferir, que é $\geq q$). Apresente uma estratégia *greedy* para resolução do problema e justifique que é correta.

5. Seja v um vetor (*array*) com n com elementos, $v[1], \ldots, v[n]$, em que cada elemento tem um valor, identificado pelo campo *key*, o qual é um inteiro positivo não superior a k. Considere o algoritmo seguinte, assumindo que os vetores de inteiros p e a têm posições suficientes.

```
ORDENA(v, n, p, k)
1. for j = 1 to k + 1
2.     a[j] = 0
3. for j = 1 to n
4.     a[1 + v[j].key] = 1 + a[1 + v[j].key]
5. for j = 3 to k
6.     a[j] = a[j] + a[j − 1]
7. for j = 1 to n
8.     p[1 + a[v[j].key]] = j
9.     a[v[j].key] = a[v[j].key] + 1
```

a) Prove que ORDENA determina em p uma permutação de $1, \ldots, n$ que define uma ordenação dos elementos de v por ordem crescente de valor (*key*). Para isso, **enuncie** e **justifique** (sucintamente):

- a condição que $a[j]$ satisfaz após a execução do bloco de instruções 1.–6.
- o invariante do bloco 7.–9., envolvendo j, p, a e v, e que conduz à correção do método.

b) Apresente a noção de algoritmo de ordenação *estável* e classifique a função ORDENA quanto à estabilidade. Justifique.

c) Caraterize a complexidade temporal assintótica de ORDENA, como função de n e k (explique sucintamente). Em que condições recomendaria este método em alternativa a MERGESORT?

(Continua, v.p.f.)

6. Recordando que o problema de decidir a existência de um ciclo hamiltoneano num grafo não dirigido pertence à classe NPC (isto é, é NP-completo), o que se pode concluir sobre o Problema do Caixeiro Viajante (**TSP**), num grafo não dirigido $G = (V, E, p)$ completo, com $p(e) \in \mathbb{Z}^+$? Justifique.

7. Considere a função MyPartition assim definida, admitindo que x é um array de n inteiros, sendo $x[1], \ldots, x[n]$ **todos distintos**, e que a e b são inteiros tais que $1 \leq a \leq b \leq n$.

```
MyPartition(x, a, b)
  z = x[b]
  j = a − 1
  for i = a to b − 1
      if x[i] > z then
          Exchange x[i] with x[j + 1]
          j = j + 1
  Exchange x[b] with x[j + 1]
  return j + 1
```

a) Apresente o resultado da chamada de MyPartition(x, a, b), para $x = [-7, 10, 23, 5, 1, -10, 15, 6]$, com $n = 8$, $a = 3$, e $b = 7$.

Nas alíneas seguintes, considere o caso geral.

b) Caraterize o estado de x e o valor de retorno após a execução de MyPartition(x, a, b).

c) Adaptando a ideia de QuickSelect, mas sem aleatorização, defina uma função recursiva MySelect$(x, 1, n, p)$ para seleção do p-**ésimo elemento menor** de x, com tempo médio de ordem $O(n)$, sendo p dado, com $1 \leq p \leq n$. A função deve usar MyPartition e pode alterar x.

d) Na continuação da alínea anterior:

1. Justifique sucintamente a correção da chamada à função MySelect$(x, 1, n, p)$, começando por ilustrar a sua aplicação à instância definida na alínea 7a) com $p = 3$ (para tal instância, deve indicar o resultado dos passos intermédios principais);
2. Apresente a expressão que define o número de comparações de MySelect no caso médio e, por aplicação do método de substituição, prove que é $O(n)$, considerando que qualquer permutação dos n valores dados seria igualmente provável.

Resolva apenas UM dos dois problemas seguintes

8. Que propriedade geométrica é explorada pelo algoritmo de Shamos para resolução do problema de determinação do par de pontos mais próximos em $O(n \log_2 n)$? Em que fase é relevante?

9. Dado um grafo não dirigido $G = (V, E)$, considere o problema de otimização linear assim definido, com variáveis de decisão x_v, para $v \in V$.

$$\text{minimizar} \sum_{v \in V} x_v$$
$$\text{sujeito a} \quad \begin{cases} x_u + x_v \geq 1, \text{ para todo } (u, v) \in E \\ 0 \leq x_u \leq 1, \text{ para todo } u \in V \end{cases}$$

Seja $C = \{v \mid x_v^\star \geq 1/2\}$, onde $x^\star$ representa uma solução ótima (sabe-se que pode ser calculada por um algoritmo polinomial). Mostre que C cobre todas as arestas de G e, recordando o problema Minimum Vertex Cover, prove que $|C|$ não excede o dobro do número de vértices de qualquer cobertura ótima.

(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

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