Departamento de Ciência de Computadores FCUP

Algoritmos (CC4010) 2016/17

1ºTeste (15.11.2016) *duração: 3h*

N.º Nome

1. Considere o problema do cálculo do produto de dois inteiros positivos, x e y, representados numa base B (fixa) com n dígitos, sendo n potência de 2. Usando uma estratégia de divisão e conquista (*divide-and-conquer*), se $n > 1$, usa-se o facto de

$$xy = (B^{n/2}x_2 + x_1)(B^{n/2}y_2 + y_1) = B^n(x_2y_2) + B^{n/2}(x_2y_1 + x_1y_2) + (x_1y_1)$$

para obter a representação de xy a partir das representações de x_1y_1, x_2y_2, x_2y_1, e x_1y_2, sendo x_1, x_2, y_1 e y_2 inteiros com $n/2$ dígitos base B. Estes produtos calculam-se pelo mesmo método (ou diretamente se $n = 1$). Os produtos por $B^{n/2}$ e B^n são efetuados por deslocamento à esquerda (*left shift*).

a) Apresente a recorrência que define a complexidade deste algoritmo e resolva-a usando o Master Theorem. Justifique a resposta sucintamente.

b) O algoritmo de Karatsuba usa o facto de $x_2y_1 + x_1y_2 = (x_1 + x_2)(y_1 + y_2) - x_1y_1 - x_2y_2$. Com que objetivo?

2. Suponha que se alterou o valor de $x[1]$ numa *heap* de máximo $x[1], \ldots, x[n]$, com n elementos, e é necessário repor a propriedade de *heap*. Para obter um majorante para a complexidade assintótica dessa operação, usou-se a recorrência $T(n) \leq T(2n/3) + c$, com c constante. Qual é a ideia que a suporta?

3. Considere a função MYPARTITION assim definida, admitindo que x é um array de n inteiros, sendo $x[1], \ldots, x[n]$, 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 $x = [-7, 10, 23, -15, 7, 4, 1, 9]$, com $a = 2$ e $b = 6$.

b) No caso geral, qual é o estado de x e o valor de retorno após a execução MYPARTITION(x, a, b)? Qual é o invariante de ciclo (para x, j e i e z) que permite justificar essa resposta?

c) Usando pseudocódigo, escreva uma versão do algoritmo *quicksort* que use MYPARTITION.

d) O que distingue o algoritmo *quicksort* do algoritmo *randomized quicksort*? Que vantagem pode ter este último?

(Continua, v.p.f.)

4. Qual é o problema que *quickselect* resolve? Como se explica a diferença de complexidade entre *randomized quicksort* e *quickselect*?

5. Considere o problema da ordenação de um *array* de n inteiros $x[1], \ldots x[n]$, por ordem decrescente. Recordando a definição das classes $O(\cdot)$, $\Omega(\cdot)$ e $\Theta(\cdot)$ e o algoritmo de ordenação por inserção, analise a correção da afirmação: "o algoritmo de ordenação por inserção tem complexidade temporal assintótica $O(n^2)$ no pior caso e $O(n)$ no melhor caso, sendo portanto $\Omega(n^2)$."

6. Dado um *array* $a[1] \ldots a[n]$, de n inteiros distintos, *ordenado* por ordem crescente, e dado um inteiro x, pretende-se determinar k tal que $|x - a[k]| = \min_{1 \leq i \leq n} |x - a[i]|$, isto é, $a[k]$ está a distância mínima de x. Por análise da árvore de decisão, prove que qualquer algoritmo comparativo que resolva o problema tem complexidade temporal $\Omega(\log n)$, no pior caso.

7. Suponha que estavam marcadas várias reuniões para um certo dia, sendo dada a hora de início (H:00, H:15, H:30, H:45, com $H \geq 8$) 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. Pretende-se minimizar o número de reuniões a cancelar.

a) Apresente a prova de que se pode obter uma solução ótima por aplicação da estratégia seguinte: "realizar a reunião que terminaria mais cedo (se houver várias, escolher uma qualquer) e proceder de modo idêntico para seleção das restantes, descartando as reuniões que seriam incompatíveis com essa escolha".

b) Comente a correção da afirmação: "Consequentemente, o problema pode ser resolvido em tempo linear no número de reuniões previstas".

(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:

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

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

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