Primeira Prova de Análise de Algoritmos

Unirio
Professor: Guilherme Dias da Fonseca
Data: 08/10/2009
Tempo de prova: 2 horas
Permitida a consulta somente a uma folha de papel A4.

1. (20 pontos) Para cada par de expressões A, B na tabela abaixo, marque Verdadeiro ou Falso para A ser O, o, Ω, ω e Θ de B. Considere que log representa o logaritmo base 2. Uma linha da tabela será considerada errada caso algum dos itens da linha estiver errado.

	A	B	O	o	Ω	ω	Θ
(a)	$n/2$	$10n$	V	F	V	F	V
(b)	n	n^2	V	V	F	F	F
(c)	$n \log n$	n	F	F	V	V	F
(d)	$\log n$	$\ln n$	V	F	V	F	V
(e)	$\log^{100} n$	$n^{1/100}$	V	V	F	F	F
(f)	$\sqrt{\log n}$	$\log \sqrt{n}$	V	V	F	F	F
(g)	$n!$	n^n	V	V	F	F	F
(h)	$\log(n!)$	$n \log n$	V	F	V	F	V

2. (20 pontos) Determine a complexidade assintótica de $T(n)$ em cada um dos itens abaixo (considere $T(O(1)) = O(1)$).

   (a)
$$T(n) = T\left(\frac{2n}{3}\right) + 1$$

   Resposta: $T(n) = \Theta(\log n)$ via teorema mestre. Caso 2 com $k = 0$ e $\log_{3/2} 1 = 0$.

   (b)
$$T(n) = \sum_{i=1}^{n} i$$

   Resposta: Progressão aritmética:

$$T(n) = n\frac{n+1}{2} = \Theta(n^2)$$

3. (20 pontos) Considere um polígono convexo P com n vértices e um ponto p. Descreva e analise a complexidade de um algoritmo sublinear (ou seja, que leve tempo $o(n)$) para determinar se o ponto p está no interior do polígono P.

   Resposta: A solução procede por busca binária. O caso base é quando o polígono tem 3 vértices, aonde decidimos em tempo constante se o ponto está dentro ou fora do triângulo. Caso tenhamos $n > 3$ vértices examinamos um par de vértices u, v tal $(n-2)/2$ vértices estejam acima da reta uv e $(n-2)/2$ vértices estejam abaixo da reta uv (se n for ímpar, deixa-se um lado com um vértice a mais). Note que se o ponto p está estiver abaixo de uv, então somente a metade do polígono abaixo de uv pode conter p. Analogamente, se o ponto p está estiver acima de uv, então somente a metade do polígono acima de uv pode conter p. Sendo assim, após tempo constante resolvemos o problema recursivamente para a metade apropriada do polígono, obtendo a recorrência a seguir para a complexidade de tempo.

$$T(n) = T\left(\frac{n}{2}\right) + 1 = O(\log n)$$

4. (20 pontos) Descreva um algoritmo de **divisão e conquista** para determinar o menor elemento e o maior elemento de um conjunto $S = \{s_1, s_2 \ldots, s_n\}$. Calcule **exatamente** o número de comparações efetuadas pelo algoritmo. O algoritmo será considerado **tão melhor quanto menor** for este número de comparações!

   Resposta: Consideramos n potência de 2. Seja $T(n)$ o número exato de comparações para n elementos na entrada. Como caso base usamos $n = 2$, aonde é possível determinar o máximo e o mínimo com somente 1 comparação. Portanto, $T(2) = 1$. Para $n > 2$, dividimos a lista em duas com $n/2$ elementos e recursivamante determinamos o máximo e mínimo de cada uma das duas listas. Combinamos as soluções com duas comparações, uma entre os dois máximos e outra entre os dois mínimos, retornando o máximo como o maior dos máximos e o mínimo como o menor dos mínimos. Portanto, $T(n) = 2T(n/2)+2$.

   Para resolvermos exatamente esta recorrência, podemos pensar na árvore obtida pelo execução do algoritmo. No nível mais baixo, são executadas $n/2$ comparações, uma para cada um dos $n/2$ pares disjuntos de elementos. Nos níveis acima, são executadas $2(n/2 - 1) = n - 2$ comparações. Portanto, temos no total $T(n) = 3n/2 - 2$ comparações.

   Para conferir, usamos indução. No caso base, verificamos que $T(2) = 1$. No passo da indução, fazemos

$$T(n) = 2\left(\frac{3n}{4} - 2\right) + 2 = \frac{3n}{2} - 2.$$

5. (20 pontos) Mostre a tabela obtida pela execução do algoritmo vistos em sala para determinar a distância de edição da palavra `MAGICA` (sem acento) para a palavra `MASCAR`. Quantas e quais edições foram usadas pelo algoritmo para chegar da primeira palavra à segunda palavra.

   Resposta:

		M	A	G	I	C	A
	0	1	2	3	4	5	6
M	1	↘**0**	→1	→2	→3	→4	→5
A	2	↓1	↘**0**	→1	→2	→3	→4
S	3	↓2	↓1	↘**1**	→**2**	→3	→4
C	4	↓3	↓2	↓2	↘2	↘**2**	→3
A	5	↓4	↓3	↓3	↓3	↓3	↘**2**
R	6	↓5	↓4	↓4	↓4	↓4	↓**3**

Foram necessárias 3 edições: substituir o G por S (↘), remover o I (→) e inserir um R no final (↓).