MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2005

Lista 3

1. Seja $T(n)$ definida pela recorrência

$$\begin{aligned} T(0) &= 1 \\ T(1) &= 1 \\ T(n) &= \max_{0 \le k \le n-1} \{T(k) + T(n-k-1)\} + n \quad \text{para } n = 2, 3, 4, \ldots \end{aligned}$$

   Mostre que $T(n) \ge n^2/2$ para todo $n \ge 0$.

2. Seja $M(n)$ definida pela recorrência

$$\begin{aligned} M(0) &= 1 \\ M(1) &= 1 \\ M(n) &= \min_{0 \le k \le n-1} \{M(k) + M(n-k-1)\} + n \quad \text{para } n = 2, 3, 4, \ldots \end{aligned}$$

   Mostre que $M(n) \ge (n+1)\lg(n+1)$ para todo $n \ge 1$.

3. Qual é o consumo de espaço do QUICKSORT no pior caso?

4. Quando um algoritmo recursivo tem como último comando executado, em algum de seus casos, uma chamada recursiva, tal chamada é denominada *recursão de calda* (*tail recursion*). Um exemplo de recursão de calda acontece no QUICKSORT.

   Toda recursão de calda pode ser substituída por uma repetição. No caso do QUICKSORT, obtemos o seguinte:

```
QUICKSORT (A, p, r)
enquanto p < r
        q ← PARTICIONE (A, p, r)
        QUICKSORT (A, p, q − 1)
        p ← q + 1
```

   Mostre como essa idéia pode ser usada (de uma maneira mais esperta) para melhorar significativamente o consumo de espaço no pior caso do QUICKSORT.

5. Quantas vezes a comparação "$A[r] \neq 0$" é executada? Defina esse número por meio de um recorrência.

```
LIMPA (A, p, r)
se p ≥ r
      então devolva r
      senão q ← LIMPA (A, p, r − 1)
            se A[r] ≠ 0
                  então q ← q + 1
                        A[q] ← A[r]
            devolva q
```

   Dê uma fórmula exata para a função definida pela recorrência. Em que classe Θ está a função definida pela recorrência? Explique.

6. Considere o seguinte algoritmo, cujo argumento n é uma potência de 2. (O algoritmo não faz nada de útil.)

```
ALGO (n)
se n ≤ 1
    então devolva 1
para i ← 1 até 8 faça
    z ← ALGO(n/2)
    para i ← 1 até n³ faça
        z ← 0
```

(1) Seja $T(n)$ o número de vezes que a atribuição "$z \leftarrow 0$" é executada. Escreva uma recorrência que define $T(n)$.

(2) Mostre que $T(n)$ é $\Omega(n^3 \log n)$.

(3) Troque "8" por "7" no algoritmo e mostre diretamente que $T(n)$ é $O(n^3)$.

7. Considere o seguinte algoritmo recursivo, cujo argumento n é um inteiro positivo.

```
ASTERISCO (n)
se n > 0
    então ASTERISCO (n − 1)
          para i ← 1 até n
              imprima "*"
          ASTERISCO(n − 1)
```

Para um dado valor de n, quantos asteriscos serão impressos em uma chamada de ASTERISCO(n)? Justifique a sua resposta mostrando os cálculos que fez para chegar a ela.

8. Qual a diferença de consumo de tempo entre uma busca binária em um vetor com n componentes e uma uma busca binária em um vetor com n^2 componentes?

9. Suponha que os elementos do vetor $A[1 \mathinner{.\,.} n]$ são distintos dois a dois. Uma inversão *significativa* é um par (i, j) de índices tal que $i < j$ mas $A[i] > 2\,A[j]$. Escreva um algoritmo que calcule o número de inversões significativas em $A[1 \mathinner{.\,.} n]$ em tempo $O(n \lg n)$.

10. Construa um algoritmo que, dados inteiros n e k, juntamente com k listas ordenadas que em conjunto tenham n registros, produza uma única lista ordenada contendo todos os registros dessas listas (isto é, faça uma *intercalação*). O seu algoritmo deve ter complexidade $O(n \lg k)$. Note que isto se transforma em $O(n \lg n)$ no caso de n listas de 1 elemento, e em $O(n)$ se só houver uma lista (de n elementos).

11. Considere a seqüência de vetores

$$A_k[1 \mathinner{.\,.} 2^k],\ A_{k-1}[1 \mathinner{.\,.} 2^{k-1}],\ \ldots,\ A_1[1 \mathinner{.\,.} 2^1],\ A_0[1 \mathinner{.\,.} 2^0]\,.$$

Suponha que cada um dos vetores é crescente. Queremos reunir, por meio de sucessivas operações de intercalação (= *merge*), o conteúdo dos vetores $A_0, \ldots, A_k$ em um único vetor crescente $B[1 \mathinner{.\,.} n]$, onde $n = 2^{k+1} - 1$. Escreva um algoritmo que faça isso em $O(n)$ unidades de tempo. Você não precisa escrever o código da rotina INTERCALA, mas precisa dizer *o que* ela faz exatamente. Justifique.

12. Considere o seguinte algoritmo que determina o segundo maior elemento de um vetor $v[1\,..\,n]$ com $n \geq 2$ números positivos distintos.

    Algoritmo Máximo (v, n)

    ```
    1. maior ← 0
    2. segundo_maior ← 0
    3. para i ← 1 até n faça
    4.     se v[i] > maior
    5.         então segundo_maior ← maior
    6.               maior ← v[i]
    7.         senão se v[i] > segundo_maior
    8.                   então segundo_maior ← v[i]
    9. devolva segundo_maior
    ```

    Suponha que v é uma permutação de 1 a n escolhida ao acaso dentre todas as permutações de 1 a n, de acordo com a distribuição uniforme de probabilidade. Seja X o número de vezes que a variável *segundo_maior* é alterada (ou seja, o número de execuções das linhas 5 e 8 do algoritmo) numa chamada de Máximo(v, n). Note que X é uma variável aleatória. Calcule o valor esperado de X.

13. Considere o seguinte algoritmo que calcula o maior e o menor elemento de um vetor $v[1\,..\,n]$ com elementos distintos.

    Algoritmo MaiorMenor (v, n)

    ```
    1. maior ← v[1]
    2. menor ← v[1]
    3. para i ← 2 até n faça
    4.     se v[i] > maior
    5.         então maior ← v[i]
    6.         senão se v[i] < menor
    7.                   então menor ← v[i]
    8. devolva maior, menor
    ```

    Suponha que a entrada do algoritmo é uma permutação de 1 a n escolhida uniformemente dentre todas as permutações de 1 a n.

    Qual é o número esperado de comparações executadas na linha 6 do algoritmo? Qual é o número esperado de atribuições efetuadas na linha 7 do algoritmo?