

MAC 5711 - Análise de Algoritmos

Departamento de Ciência da Computação

Segundo semestre de 2025

Lista 2

1. Resolva as recorrências abaixo.

(a) $T(n) = 2T(\lfloor n/2 \rfloor) + \Theta(n^2)$

(b) $T(n) = 8T(\lfloor n/2 \rfloor) + \Theta(n^2)$

(c) $T(n) = 2T(\lfloor n/4 \rfloor) + 2$

(d) $T(n) = 2T(\lfloor n/2 \rfloor) + \Theta(n^3)$

(e) $T(n) = 7T(\lfloor n/3 \rfloor) + \Theta(n^2)$

(f) $T(n) = T(\lfloor 9n/10 \rfloor) + \Theta(n)$

2. Escreva um algoritmo que ordena uma lista de n itens dividindo-a em três sublistas de aproximadamente $n/3$ itens, ordenando cada sublista recursivamente e intercalando as três sublistas ordenadas. Analise seu algoritmo concluindo qual é o seu consumo de tempo.
3. Seja $X[1..n]$ um vetor de inteiros e i e j dois índices distintos de X , ou seja, i e j são inteiros entre 1 e n . Dizemos que o par (i, j) é uma *inversão* de X se $i < j$ e $X[i] > X[j]$. Escreva um algoritmo $O(n \lg n)$ que devolva o número de inversões em um vetor X , onde n é o número de elementos em X .
4. Descreva 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 duas listas (no total com n elementos).
5. Considere a sequência de vetores $A_k[1..2^k], A_{k-1}[1..2^{k-1}], \dots, A_1[1..2^1]$, e $A_0[1..2^0]$. Suponha que cada um dos vetores é crescente. Queremos reunir, por meio de sucessivas operações de intercalação (= *merge*), o conteúdo dos vetores $A_0, \dots, A_k$ em um único vetor crescente $B[1..n]$, onde $n = 2^{k+1} - 1$. Escreva um algoritmo que faça isso em $O(n)$ unidades de tempo. Use como subrotina o INTERCALE visto em aula.
6. Considere o seguinte algoritmo recursivo, cujo argumento n é um inteiro positivo.

ASTERISCO (n)

1 se $n > 0$

2 então ASTERISCO ($n - 1$)

3 para $i \leftarrow 1$ até n

4 imprima “*”

5 ASTERISCO($n - 1$)

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

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

```

LIMPA ( $A, p, r$ )
1   se  $p \geq r$ 
2       então devolva  $r$ 
3       senão  $q \leftarrow \text{LIMPA}(A, p, r - 1)$ 
4           se  $A[r] \neq 0$ 
5               então  $q \leftarrow q + 1$ 
6                    $A[q] \leftarrow A[r]$ 
7           devolva  $q$ 

```

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

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

```

ALGO ( $n$ )
1   se  $n \leq 1$ 
2       então devolva 1
3   para  $i \leftarrow 1$  até 8 faça
4        $z \leftarrow \text{ALGO}(n/2)$ 
5       para  $i \leftarrow 1$  até  $n^3$  faça
6            $z \leftarrow 0$ 

```

- (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 $O(n^3)$.
9. Qual a diferença de consumo de tempo entre uma busca binária em um vetor com n componentes e uma busca binária em um vetor com n^2 componentes?