

- 1) Considere como entrada um conjunto S com n números reais e, um número real x :
- Projete um algoritmo para determinar se há dois elementos de S cuja soma seja exatamente x . O algoritmo deverá rodar em tempo proporcional a $O(n \lg n)$.
 - Suponha agora que o conjunto S está ordenado. Projete um novo algoritmo que resolva o problema anterior em tempo proporcional a $O(n)$.

- 2) Três “pesquisadores” propuseram o seguinte algoritmo de ordenação “elegante”:

```

STOOGE-SORT( $A, i, j$ )
1  se  $A[i] > A[j]$ 
2    então troca( $A[i], A[j]$ )
3  se  $i + 1 \geq j$ 
4    então “finaliza processamento”
5   $k \leftarrow \lfloor (j-i+1)/3 \rfloor$            {Arredonda para menos}
6  STOOGE-SORT( $A, i, j-k$ )         {primeiros dois terços}
7  STOOGE-SORT( $A, i+k, j$ )         {(últimos dois terços)}
8  STOOGE-SORT( $A, i, j-k$ )         {primeiros dois terços novamente}

```

a) Mostre que para n elementos, **STOOGE-SORT**($A, 1, n$) ordena corretamente o arranjo de entrada $A[1..n]$.

b) Forneça uma recorrência para o tempo de execução no pior caso de **STOOGE-SORT** e um limite assintótico restrito (notação Θ) sobre o tempo de execução no pior caso.

- 3) Desenhe a árvore de recursão para o algoritmo Mergesort aplicado a um vetor de 16 elementos. Por que a técnica de programação dinâmica não é capaz de acelerar o algoritmo?

- 4) Sabe-se que, para $m > 1$ e $n > 1$:

$$\binom{m}{n} = \begin{cases} \binom{m-1}{n} + \binom{m-1}{n-1} & \text{se } n > 0 \text{ e } m > n \\ 1 & \text{se } n = 0 \text{ ou } m = n \end{cases},$$

Considere as duas maneiras de implementar uma função que calcula, dados m e n , o valor de $\binom{m}{n}$:

```

A( $m, n$ )
1. se  $n = 0$  ou  $m = n$ 
2.   então devolve 1
3.   senão devolve  $A(m-1, n) + A(m-1, n-1)$ 

```

```

B( $m, n$ )
1. para  $i \leftarrow 0$  até  $m$ 
2.    $t[i, 0] \leftarrow 1$ 
3.    $t[i, i] \leftarrow 1$ 
4. para  $i \leftarrow 2$  até  $m$ 
5.   para  $j \leftarrow 1$  até  $i-1$ 
6.      $t[i, j] \leftarrow t[i-1, j] + t[i-1, j-1]$ 
7. devolve  $t[m, n]$ 

```

- Qual é a complexidade de cada uma das funções? Justifique sua resposta.
- Qual é a mais eficiente? Por quê?