

1) 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?

2) 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?

#### Ex3 A maior soma na matriz de números

Dada uma matriz de números inteiros positivos e negativos, escreva e implemente um algoritmo que encontre um retângulo com a maior soma contido na matriz. A soma de um retângulo é a soma de todos os elementos dentro do retângulo.

Neste problema, o retângulo com a maior soma é referido como maximal. Um retângulo é qualquer submatriz ( $n \times m$ ),  $1 \leq n, m \leq N$ , localizado dentro da matriz.

Como exemplo, o retângulo maximal da matriz:

0	-2	-7	0	Será			Com soma = 15
9	2	-6	2		9	2	
-4	1	-4	1		-4	1	
-1	8	0	-2		-1	8	