# Cálculo tempo execução - Insertion Sort

Gabriella Silvestre Annunciato - Fatec Antônio Russo - São Caetano do Sul

# Cálculo tempo execução - Insertion Sort

Linha 1: for j ← 2 to length[A]

* Executa n - 1 vezes
* Cada iteração: 1 atribuição (j ← ...) + 1 teste lógico (j ≤ n)
* Total: (n - 1) iterações × 2t = 2t(n - 1)
* Mais 1 teste lógico extra ao final do laço → + t

Linha 2: key ← A[j]

* 1 acesso + 1 atribuição → 2t por iteração
* Total: 2t(n - 1)

Linha 4: i ← j - 1

* 1 aritmética (j - 1) + 1 atribuição → 2t por iteração
* Total: 2t(n - 1)

Linha 5: while i > 0 and A[i] > key

Em média no pior caso, roda até j - 1 vezes para cada j

* Cada comparação:
* i > 0 → 1 comparação
* A[i] > key → 1 acesso + 1 comparação
* Total: 3t por iteração do while
* Soma total no pior caso:

Linha 6: A[i + 1] ← A[i]

* 1 aritmética (i + 1) + 2 acessos + 1 atribuição → 4t
* Executada no mesmo número de vezes que o while:

Linha 7: i ← i - 1

* 1 aritmética + 1 atribuição → 2t
* Mesmo número de vezes:

Linha 8: A[i + 1] ← key

* 1 aritmética + 1 atribuição → 2t
* Executada n - 1 vezes
* Total: 2t(n - 1)

# Total:

Somando todas as parcelas:

Termos lineares (laço externo):

● 2t(n - 1) (linha 1)

● 2t(n - 1) (linha 2)

● 2t(n - 1) (linha 4)

● 2t(n - 1) (linha 8)

● +t (teste final do for)

Subtotal: (2t + 2t + 2t + 2t)(n − 1) + t = 8t(n − 1) + t

Termos quadráticos (laço interno):

* Comparações do while:
* Atribuições A[i+1] ← A[i]:
* Decremento i ← i - 1:

Subtotal:

# Expressão Final do Tempo de Execução