AULA 2

Introdução à AA

CLRS 2.1–2.2
AU 3.3, 3.6

Exercício 2.A

Quanto vale S no fim do algoritmo?

```
1  S ← 0
2  para i ← 2 até n − 2 faça
3      para j ← i até n faça
4          S ← S + 1
```

Escreva um algoritmo mais eficiente
que tenha o mesmo efeito.

Solução

Se $n \geq 4$ então no fim da execução das linhas 1–4,

$$\begin{aligned} S &= (n-1) + (n-2) + \cdots + 4 + 3 \\ &= (n-3)(n+2)/2 \\ &= \tfrac{1}{2}n^2 - \tfrac{1}{2}n - 3\,. \end{aligned}$$

Análise da ordenação por inserção

Algoritmo rearranja $A[1\,..\,n]$ em ordem crescente
($A[1\,..\,n]$ é *crescente* se $A[1] \leq \cdots \leq A[n]$)

```
ORDENA-POR-INSERÇÃO (A, n)
1  para j ← 2 até n faça
2      chave ← A[j]
3      i ← j − 1
4      enquanto i ≥ 1 e A[i] > chave faça
5          A[i + 1] ← A[i]
6          i ← i − 1
7      A[i + 1] ← chave
```

O algoritmo faz o que promete?

Invariante: antes de cada execução do bloco de linhas 2–7, $A[1\,..\,j{-}1]$ é crescente

1						j				n
22	33	33	33	44	55	11	99	22	55	77

Prove!

Quantas atribuições ($\leftarrow$) algoritmo faz?

3 $\quad i \leftarrow j - 1 \qquad \triangleright \quad 2 \leq j \leq n$

4 $\quad$ enquanto $i \geq 1$ e $A[i] > chave$ faça

5 $\qquad A[i+1] \leftarrow A[i]$

6 $\qquad i \leftarrow i - 1$

linha	atribuições
3	1
4	0
5	$\leq j - 1 \leq n - 1$
6	$\leq n - 1$
3–6	$\leq 2n - 1$

1 para $j \leftarrow 2$ até n faça $\triangleright$ $j \leftarrow j + 1$ escondido

2 $\quad chave \leftarrow A[j]$

3 $\quad$ LINHAS 3–6

7 $\quad A[i+1] \leftarrow chave$

linha	atribuições
1	$n - 1 + 1$
2	$n - 1$
3–6	$\leq (n-1)(2n-1)$
7	$n - 1$
total	$\leq 2n^2 - 1$

Análise mais fina do número de atribuições:

linha	atribuições
1	$n-1+1$
2	$n-1$
3	$n-1$
4	0
5	$\leq 1+2+\cdots+(n{-}1) = n(n-1)/2$
6	$\leq 1+2+\cdots+(n{-}1) = n(n-1)/2$
7	$n-1$
total	$\leq n^2+3n-3$

n	n^2+3n-3	n^2
1	1	1
2	7	4
3	15	9
10	127	100
100	10297	10000
1000	1002997	1000000
10000	100029997	100000000
100000	10000299997	10000000000

n^2 domina os outros termos