# Complexité



Selection Sort (A, n) : — Taille du pb — tableau de n entier

| $i+1$ petit éléments triés | $n-i$ + gds éléments non triés |
|---|---|

```
for i ← 0 to n-2
    posmin ← i
    for j ← i+1 to n-1
        if A[j] < A[posmin]
            posmin ← j
        endif
    end for
    A[posmin] ↔ A[i]
end for
```

| coût | nbre exec |
|---|---|
| $c_1$ | $n-1$ |
| $c_2$ | $n-1$ |
| $c_3$ | $n(n-1)/2$ |
| $c_4$ | $n(n-1)/2$ |
| $c_5$ | $\leq n(n-1)/2$ |
| $c_6$ | $n-1$ |

$$\sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} n-i-1 = \frac{(\overbrace{n-1}^{pt} + \overbrace{1}^{dt})(\overbrace{n-1}^{nt})}{2} = n(n-1)/2$$

Equation : $\underbrace{(n-1)(c_1+c_2+c_6) + \frac{n(n-1)}{2}(c_3+c_4)}_{an^2+bn+c} \leq T(n) \leq (n-1)(c_1+c_2+c_6) + \frac{n(n-1)}{2}(c_3+c_4+c_5)$

$an^2+bn+c \leq T(n) \leq a'n^2+b'n+c'$

La complexité est quadratique dans tous les cas

=> On peut faire des expériences si on veut trouver les coefs.

InsertionSort (A, n) :

```
for i ← 1 to n-1
    key ← A[i]
    j ← i-1
    while j >= 0 and A[j] > key
        A[j+1] ← A[j]
        j ← j-1
    end while
    A[j+1] ← key
endfor
```

Soit $t_i$ le nbre d'exécution du corps while pour un i donné

| # exec |
|---|
| $n-1$ |
| $n-1$ |
| $n-1$ |
| $\sum_{i=1}^{n-1} t_i+1$ |
| $\sum_{i=1}^{n-1} t_i$ |
| $\sum_{i=1}^{n-1} t_i$ |
| $n-1$ |

$\sum t_i = 0$

$\sum_{i=1}^{n-1} t_i+1 = n-1$

$T(n)$ est de la forme $an+b$

=> Linéaire

Dans le pire des cas :

$t_i = i$ (tableau à l'envers)

$\sum_{i=1}^{n-1} t_i = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$

$\sum_{i=1}^{n-1} (t_i+1) = \sum_{i=1}^{n-1} i+1 = \frac{(n+2)(n-1)}{2}$

$T(n)$ est de la forme $a'n^2+b'n+c$

=> Quadratique