

# Algo 3 : Etude d'une tâche

## Trier les valeurs d'un tableau

**Algorithme 5** : `tri_insertion` ( $A, n$ )

**Entrées** :

- $A$  : tableau a priori non trié
- $n$  : nb d'éléments dans  $A$

**Sorties** :

- Le tableau  $A$  avec ses éléments triés par ordre croissant

**Procédure** :

1. Pour  $i$  de 2 à  $n$  :
  - A. Fixer *clé* à  $A[i]$  et  $j$  à  $i-1$
  - B. Tant que  $j > 0$  et  $A[j] > clé$ , faire la chose suivante :
    - i. Fixer  $A[j+1]$  à  $A[j]$
    - ii. Décrémenter  $j$  (soit fixer  $j$  à  $j - 1$ )
  - C. Fixer  $A[j+1]$  à *clé*

### Exercice 1

On considère le tableau  $A = [10, 4, 12, -2, 4]$

(le même tableau que dans la première séance d'algorithmique).

(1) Appliquer l'algorithme `tri_insertion` sur ce tableau. On fera le résumé de cet algorithme par une série de schéma sur les différentes étapes du tableau  $A$ .

(2) Pourquoi parle-t-on de tri par insertion ?

(3) Comment évolue le taux de croissance du temps d'exécution du tri insertion. On utilisera la notation  $\mathcal{O}()$ .

**Algorithme 6** : `tri_sélection` ( $A, n$ )

**Entrées** :

- $A$  : tableau a priori non trié
- $n$  : nb d'éléments dans  $A$

**Sorties** :

- Le tableau  $A$  avec ses éléments triés par ordre croissant

**Procédure** :

1. Pour  $i$  de 1 à  $n - 1$  :
  - A. Fixer *minimum* à l'indice du plus petit élément du sous-tableau  $A[i..n]$
  - B. Echanger  $A[i]$  et  $A[\text{minimum}]$

### Exercice 2

On considère le tableau  $A = [10, 4, 12, -2, 4]$  (toujours le même tableau).

(1) Appliquer l'algorithme `tri_sélection` sur ce tableau. On fera le résumé de cet algorithme par une série de schéma sur les différentes étapes du tableau  $A$ .

(2) Pourquoi parle-t-on de tri par sélection ?

(3) Comment évolue le taux de croissance du temps d'exécution du tri sélection. On utilisera la notation  $\mathcal{O}()$ .