

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

Terminaison

Correction

Complexité

# Tris

Christophe Viroulaud

Première - NSI

**Algo 04**



FIGURE 1 – Trier un jeu de cartes est un problème informatique.

## Algorithmes de tris

- Recherche
- Tri par sélection
- Tri par insertion

## Implémentation

- Rappel : Passer un tableau à une fonction
- Implémentations des tris

## Études des implémentations

- Terminaison
- Correction
- Complexité

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

## Études des implémentations

Terminaison

Correction

Complexité

Déterminer plusieurs méthodes de tris de données.

## 1. Algorithmes de tris

### 1.1 Recherche

### 1.2 Tri par sélection

### 1.3 Tri par insertion

## 2. Implémentation

## 3. Études des implémentations

### Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

### Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

### Études des implémentations

Terminaison

Correction

Complexité



## Algorithmes de tris

### Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

## Études des implémentations

Terminaison

Correction

Complexité

### Activité 1 :

1. Prendre le paquet de cartes mélangées et les étaler sur la table.
2. Trier les cartes.
3. Formaliser la méthode utilisée sous forme d'un algorithme.

## 1. Algorithmes de tris

### 1.1 Recherche

### 1.2 Tri par sélection

### 1.3 Tri par insertion

## 2. Implémentation

## 3. Études des implémentations

### Algorithmes de tris

Recherche

**Tri par sélection**

Tri par insertion

### Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

### Études des implémentations

Terminaison

Correction

Complexité

## Algorithmes de tris

Recherche

**Tri par sélection**

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

## Études des implémentations

Terminaison

Correction

Complexité

- ▶ Pour chaque carte du tas :
  - ▶ Trouver la plus petite carte dans la partie non triée.
  - ▶ Échanger cette carte avec la première de la partie non triée.

## Algorithmes de tris

Recherche

**Tri par sélection**

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

## Études des implémentations

Terminaison

Correction

Complexité



6	5	10	12	4	2	11	13	1	3	7	9	8
---	---	----	----	---	---	----	----	---	---	---	---	---

FIGURE 2 – Modélisation



## Algorithmes de tris

Recherche

**Tri par sélection**

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

Terminaison

Correction

Complexité

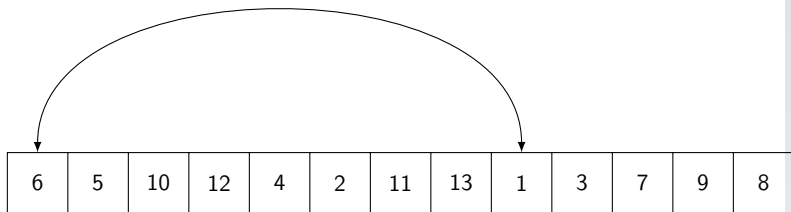


FIGURE 3 – Sélection du plus petit élément.

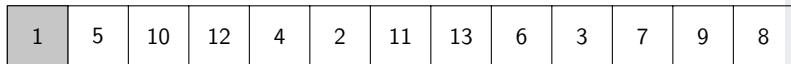


FIGURE 4 – La partie triée est à gauche.

## Algorithmes de tris

Recherche

**Tri par sélection**

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

Terminaison

Correction

Complexité

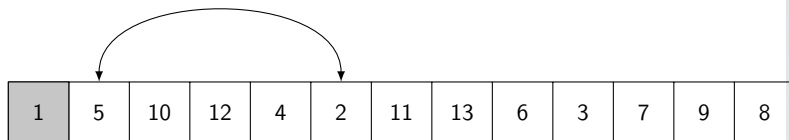


FIGURE 5 – Sélection du plus petit élément.

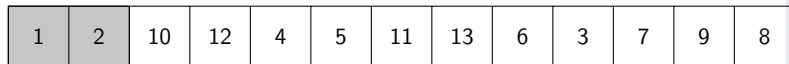


FIGURE 6 – La partie triée est à gauche.

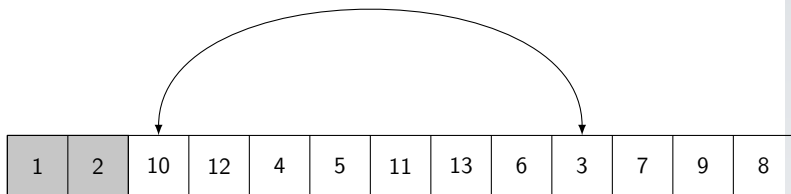


FIGURE 7 – Sélection du plus petit élément.

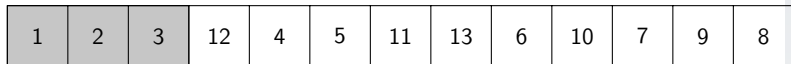


FIGURE 8 – La partie triée est à gauche.

## 1. Algorithmes de tris

### 1.1 Recherche

### 1.2 Tri par sélection

### 1.3 Tri par insertion

## 2. Implémentation

## 3. Études des implémentations

### Algorithmes de tris

Recherche

Tri par sélection

**Tri par insertion**

### Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

### Études des implémentations

Terminaison

Correction

Complexité

## Algorithmes de tris

Recherche

Tri par sélection

**Tri par insertion**

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

## Études des implémentations

Terminaison

Correction

Complexité

- ▶ Pour chaque carte du tas :
  - ▶ Tant que la carte précédente est plus petite
  - ▶ Échanger cette carte avec la carte en cours.



6	5	10	12	4	2	11	13	1	3	7	9	8
---	---	----	----	---	---	----	----	---	---	---	---	---

FIGURE 9 – Modélisation

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

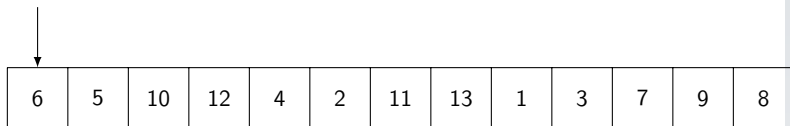
Implémentations des tris

## Études des implémentations

Terminaison

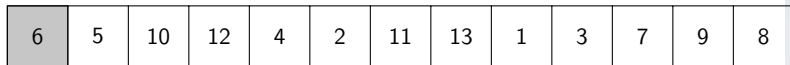
Correction

Complexité



6	5	10	12	4	2	11	13	1	3	7	9	8
---	---	----	----	---	---	----	----	---	---	---	---	---

FIGURE 10 – Carte en cours



6	5	10	12	4	2	11	13	1	3	7	9	8
---	---	----	----	---	---	----	----	---	---	---	---	---

FIGURE 11 – La partie triée est à gauche.

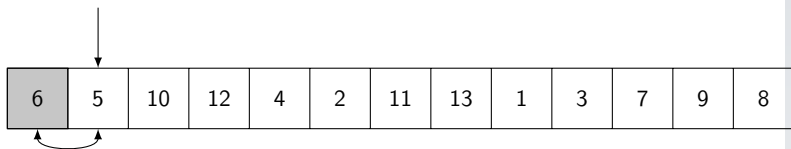


FIGURE 12 – Carte en cours

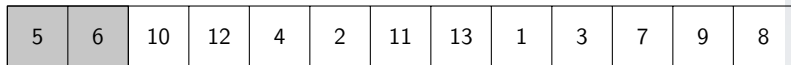


FIGURE 13 – La partie triée est à gauche.



## Algorithmes de tris

Recherche

Tri par sélection

**Tri par insertion**

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

Terminaison

Correction

Complexité

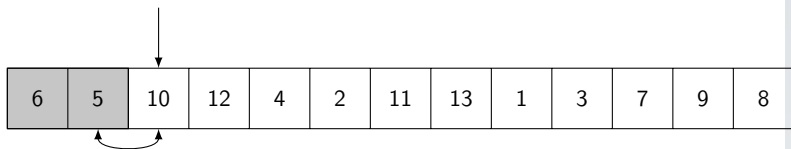


FIGURE 14 – Carte en cours

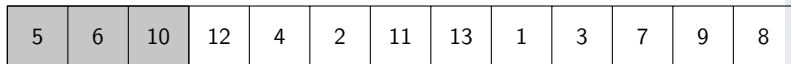


FIGURE 15 – La partie triée est à gauche.

## Algorithmes de tris

Recherche

Tri par sélection

**Tri par insertion**

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

Terminaison

Correction

Complexité

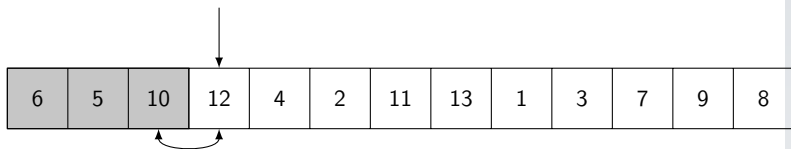


FIGURE 16 – Carte en cours

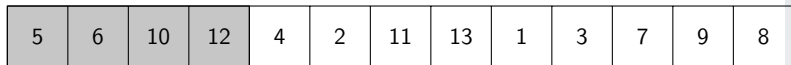


FIGURE 17 – La partie triée est à gauche.

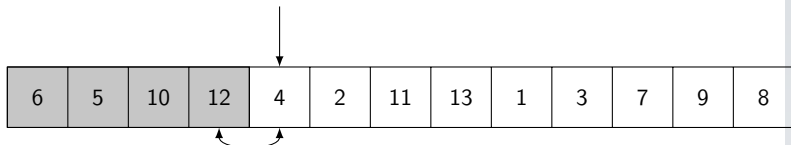


FIGURE 18 – Carte en cours

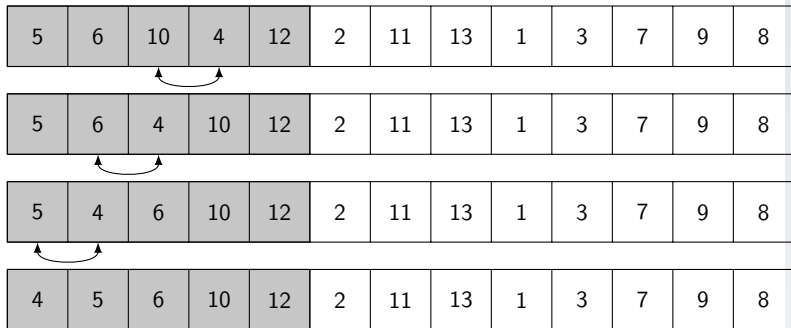


FIGURE 19 – La partie triée est à gauche.

Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

Terminaison

Correction

Complexité

## 1. Algorithmes de tris

## 2. Implémentation

### 2.1 Rappel : Passer un tableau à une fonction

### 2.2 Implémentations des tris

## 3. Études des implémentations

### Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

### Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

### Études des implémentations

Terminaison

Correction

Complexité

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

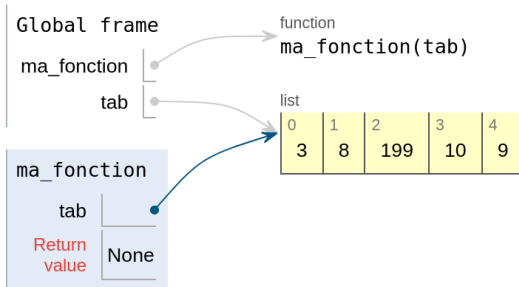
Terminaison

Correction

Complexité

```
1 def ma_fonction(tab: list) -> None:
2     tab[2] = 199
3
4     tab = [3, 8, 1, 10, 9]
5     ma_fonction(tab)
```

Code 1 – Quand on passe un tableau en argument à une fonction, on passe en réalité **une référence** au tableau original.



## 1. Algorithmes de tris

## 2. Implémentation

### 2.1 Rappel : Passer un tableau à une fonction

### 2.2 Implémentations des tris

## 3. Études des implémentations

### Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

### Implémentation

Rappel : Passer un tableau  
à une fonction

**Implémentations des tris**

### Études des implémentations

Terminaison

Correction

Complexité

# Implémentation des tris

- Pour chaque carte du tas :
  - **Trouver la plus petite** carte dans la partie non triée.
  - **Échanger** cette carte avec la première de la partie non triée.

## Activité 2 :

1. Écrire la fonction `indice_mini(tab: list, dep: int) → int` qui renvoie l'indice de la valeur minimale de `tab`, entre l'élément d'indice `deb` et la fin du tableau.
2. Écrire la fonction `echanger(tab: list, i: int, j: int) → None` qui échange les éléments d'indices `i` et `j`.
3. Écrire alors la fonction `tri_insertion(tab: list) → None`.

Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

Terminaison

Correction

Complexité

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

## Études des implémentations

Terminaison

Correction

Complexité

```
1 def indice_mini(tab: list, dep: int) -> int:
2     i_mini = dep
3     mini = tab[dep]
4     # parcours de la partie du tableau
5     for i in range(dep, len(tab)):
6         if tab[i] < mini:
7             i_mini = i
8             mini = tab[i]
9     return i_mini
```



## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

**Implémentations des tris**

## Études des implémentations

Terminaison

Correction

Complexité

```
1 def echanger(tab: list, i: int, j: int) -> None:
2     temp = tab[i]
3     tab[i] = tab[j]
4     tab[j] = temp
```

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

**Implémentations des tris**

## Études des implémentations

Terminaison

Correction

Complexité

```
1 def tri_selection(tab: list) -> None:
2     for i in range(len(tab)):
3         i_mini = indice_mini(tab, i)
4         echanger(tab, i, i_mini)
```

- Pour chaque carte du tas :
  - Tant que la carte précédente est plus petite
  - **Échanger** cette carte avec la carte en cours.

### Activité 3 :

1. Écrire la fonction `insérer(tab: list, j: int)`  
→ `None` qui insère l'élément de rang `j` dans la partie déjà triée.
2. Écrire alors la fonction `tri_insertion(tab: list)` → `None`.

#### Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

#### Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

#### Études des implémentations

Terminaison

Correction

Complexité

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

## Études des implémentations

Terminaison

Correction

Complexité

```
1 def inserer(tab: list, j: int) -> None:
2     while j-1 >= 0 and tab[j-1] > tab[j]:
3         echanger(tab, j-1, j)
4         j = j-1
```

## Remarque

La condition  $j-1 \geq 0$  évite de *sortir* du tableau.

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

Terminaison

Correction

Complexité

```
1 def tri_insertion(tab: list) -> None:
2     for i in range(len(tab)):
3         inserer(tab, i)
```

## 1. Algorithmes de tris

## 2. Implémentation

## 3. Études des implémentations

### 3.1 Terminaison

### 3.2 Correction

### 3.3 Complexité

#### Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

#### Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

#### Études des implémentations

Terminaison

Correction

Complexité

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

## Études des implémentations

### Terminaison

Correction

Complexité

## À retenir

Pour montrer que l'algorithme termine (ne part pas dans une boucle sans fin), on utilise **un variant de boucle**.

## Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

## Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

Études des  
implémentations

## Terminaison

Correction

Complexité

Dans le tri par sélection, on utilise deux boucles **bornées**.  
Dans chaque boucle, le variant utilisé est la  $v$



## 1. Algorithmes de tris

## 2. Implémentation

## 3. Études des implémentations

### 3.1 Terminaison

### 3.2 Correction

### 3.3 Complexité

#### Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

#### Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

#### Études des implémentations

Terminaison

**Correction**

Complexité

## 1. Algorithmes de tris

## 2. Implémentation

## 3. Études des implémentations

### 3.1 Terminaison

### 3.2 Correction

### 3.3 Complexité

#### Algorithmes de tris

Recherche

Tri par sélection

Tri par insertion

#### Implémentation

Rappel : Passer un tableau  
à une fonction

Implémentations des tris

#### Études des implémentations

Terminaison

Correction

**Complexité**