

# Algorithmique

Laurent Debize



# Tri par insertion

C'est le tri du joueur de cartes : son éventail à la main, il considère la carte de gauche comme déjà triée, place la seconde par rapport à la première, insère la troisième par rapport aux deux premières.

C'est humainement rapide parce qu'il n'y a pas (ou peu) de problème de coulissage de cartes. Informatiquement, c'est cela qui prend du temps.

# Tri par insertion

On retire un nombre à la position  $i$ .

Les  $i-1$  premiers éléments de  $A$  sont déjà triés.

Tant que l'on n'est pas arrivé au début du tableau,

et que l'élément courant est plus grand que celui à insérer,

On décale l'élément courant (on le met dans la place vide).

On s'intéresse à l'élément précédent.

Finalement, on a trouvé où insérer notre nombre.

# Tri par insertion

---

**Fonction** TriParInsertion( $A[0..n-1]$  : flottants)

---

**Variables locales :**

i, k : entiers; element : flottant

**début**

```
    pour i de 1 à n-1 faire
        element ← A[i]
        k ← i - 1
        tant que element < A[k] et k ≥ 0 faire
            A[k+1] ← A[k]
            k ← k - 1
        fin
        A[k+1] ← element
    fin
fin
```

---

## Exercice 1

Vous répondrez à toutes les questions dans un document Word que vous m'enverrez par mail ensuite en précisant dans l'objet : **TIIS2**.

- 1 Implémenter cet algorithme et le tester sur un tableau quelconque.
- 2 On se propose d'estimer la complexité de cet algorithme. Pour cela, on va compter le nombre d'itérations.
  - Placer un compteur à l'intérieur de la boucle **tant que**
  - Afficher la valeur de ce compteur à la fin de l'algorithme.
  - Faire varier la taille  $n$  du tableau entre 1 et 10 (on pourra se servir d'une boucle qui entoure tout l'algorithme)
  - Relever la valeur du compteur pour chaque taille  $n$  du tableau. Placer ces valeurs dans un tableau Excel.
  - Tracer sur un graphique Excel la valeur du compteur en fonction de la taille du tableau
  - Faire un clic droit sur la courbe, et ajouter une courbe de tendance. Dans les options, sélectionner **Puissance**, et cocher **Afficher l'équation sur le graphique**.
  - Quelle est l'équation qui s'affiche sur le graphique ? Cette équation traduit la complexité de l'algorithme en fonction de la taille des données, ici  $n$ .

# Tri par sélection

C'est un autre tri de joueur de cartes (?) : on a l'éventail en main, on repère la plus petite, on la met à gauche, puis on repère la plus petite des restantes, on la met à droite de la précédente, etc.

# Tri par sélection

---

**Fonction** TriParSelection( $A[0..n-1]$  : flottants)

---

i, k, position\_min : entiers ;

min, stock : flottant

**début**

```
    pour i de 0 à n-1 faire
        min ← A[i]
        position_min ← i
        pour k de i à n-1 faire
            si A[k] < min alors
                min ← A[k]
                position_min ← k
            fin
        fin
        stock ← A[i]
        A[i] ← A[position_min]
        A[position_min] ← stock
    fin
fin
```

---

## Exercice 2

- Répondez pour cet algorithme aux mêmes questions que pour le tri par insertion et intégrez ces réponses dans le document à rendre.  
Vous insérerez le compteur à l'intérieur de la deuxième boucle pour (celle sur  $k$ )



# Tri bulle

On va balayer le tableau pour tous les indices de  $n-1$  à  $1$ , quand le terme  $A[k]$  est plus petit que le terme  $A[k-1]$ , on les échange. Quand on a fini cette phase, le plus petit est en première place. On recommence alors de  $n-1$  à  $2$ , puis de  $n-1$  à  $3$ , ..., de  $n-1$  à  $n-2$ .

## Tri bulle

---

**Fonction** TriBulle( $A[0..n-1]$  : flottants)

---

 $i, k$  : entiers; stock : flottant**début**

```
    pour  $i$  de 0 à  $n-1$  faire
        pour  $k$  de  $n-1$  à  $i+1$  faire
            si  $A[k] < A[k-1]$  alors
                stock  $\leftarrow A[k-1]$ 
                 $A[k-1] \leftarrow A[k]$ 
                 $A[k] \leftarrow$  stock
            fin
        fin
    fin
fin
```

---

## Exercice 3

- Répondez pour cet algorithme aux même questions que pour le tri par insertion et intégrez ces réponses dans le document à rendre.  
Vous insérerez le compteur à l'intérieur de la deuxième boucle pour (celle sur  $k$ )
- Comparez la complexité de ces trois algorithmes de tri.