6.4 tri par insertion

NSI 1ère - JB Duthoit

6.4.1 Principe

On insère un à un les éléments parmi ceux déjà trié.

☞ Le **tri par insertion** est aussi appelé "tri naturel" : c'est souvent ce tri que nous utilisons naturellement pour trier un jeu de cartes par exemple.

VIDÉO Une animation pour comprendre le tri par insertion - Cliquez ici !

Exercice 6.137

Après avoir étudié la vidéo, donner les différentes étapes du tri par insertion pour le tableau suivant : $T = [8, 5, 1, 9, 7]$

6.4.2 L'algorithme

Exercice 6.138

1. Construire un algorithme d'une fonction `inserer(t,i)` avec `i` un entier non nul , et `t` un tableau d'entiers où les valeurs de l'indice 0 à i-1 sont triées
   La fonction doit insérer la valeur t[i] "au bon endroit" afin que les valeurs de l'indice `0` à l'indice `i` soit triées.
2. Implémenter cet algorithme en python.

Exercice 6.139

Savoir Construire l'algorithme correspondant au tri par insertion

On arrive à l'algorithme suivant :

```
FONCTION tri_ insertion(T :tableau d'entiers)
    POUR i DE 1 A longueur(T) − 1 FAIRE
        inserer(t,i) insere(
```

- Implémenter votre algorithme en Python
- Écrire les pré_conditions
- Écrire les post_conditions
- Donner un jeu de tests, sous forme de plusieurs assertions

6.4.3 Complexité temporelle

Savoir-Faire 6.17

Savoir déterminer la complexité du tri par insertion

Dans le pire des cas, le tableau est rangé dans l'ordre décroissante.

Soit $T(n)$ Le nombre de comparaisons.

Calculer $T(n)$, et en déduire la complexité.

***

6.4.4 Terminaison

La boucke POUR ne pose aucun problème !
En revanche, il faut faire attention à la boucle TANT QUE qui peut ne jamais terminer !
La condition de la boucle TANT QUE "est $j < 0$ and $T[j-1] > x$".
La boucle s'arrêtte quant l'une ou l'autre des conditions n'est plus vraies.
Ici, on décrémente j à chaque fois de, donc nous sommes certains que la condition $j > 0$ ne sera plus vraie à partir d'une certaine étape !

☞ Notre algorithme s'arrêtera donc !

6.4.5 Validité de l'algorithme

Démonstration 6.2

☛ Invariant de boucle

Considérons la propriété P(i) : "Le tableau T[0],T[1],...,T[i-1] est trié ".

Étape 1 : La propriété est-elle vraie pour i=0 ? :

Étape 2 : On suppose la propriété vraie pour l'entier k. Est-elle vraie pour l'entier k+1 ?

Etape 3 : Et pour le dernier passage dans la boucle ?

Étape 4 : Conclusion