
	<b>Séquence 5</b>	<b>1 / 1</b>
	<b>Algorithme de tri</b> <b>Le tri par insertion</b>	

## 1 Objectif

Mettre en œuvre un algorithme de tri.

## 2 Principe du tri par insertion

On recherche le plus petit élément du tableau et on le met à sa place, en l'échangeant avec le premier.

On recherche le second plus petit élément (qui est donc le plus petit élément du tableau allant du 2<sup>ième</sup> élément à la fin) et on le met à sa place (donc en 2<sup>ième</sup> position) par un échange avec le 2<sup>ième</sup> élément du tableau.

Etc.

## 3 Exemple

Tableau de départ : [7, 8, 3, 5, 6, 2, 11] : le minimum du tableau est 2.

On l'échange avec le premier : [ 2 ,        ,        ,        ,        ,        ].

Le minimum du sous-tableau souligné est :

On l'échange avec le 2<sup>ième</sup> élément du tableau : [ ,        ,        ,        ,        ,        ].

Le minimum du sous-tableau souligné est :

On l'échange avec le 3<sup>ième</sup> élément du tableau : [ , ,        ,        ,        ,        ].

Le minimum du sous-tableau souligné est :

On l'échange avec le 4<sup>ième</sup> élément du tableau : [ , , ,        ,        ,        ].

Le minimum du sous-tableau souligné est :

On l'échange avec le 5<sup>ième</sup> élément du tableau : [ , , , ,        ,        ].

Le minimum du sous-tableau souligné est : et il est à sa place.

Comme il ne reste que deux cases, le tableau est trié.

## 4 Travail demandé

**Question 1.** *A la main* : **écrire** les contenus successifs du tableau (liste)  $T = [4, 1, 3, 2, 25, 12, 7, 21, 9]$  dans les différentes étapes du tri par insertion.

**Question 2.** **Ecrire** l'algorithme correspondant au tri par insertion.

**Question 3.** **Planter** l'algorithme en écrivant une fonction `tri_insertion ( T )` qui trie le tableau de nombres  $T$  par ordre croissant.

**Question 4.** **3) Tester** votre fonction sur quelques exemples.