Tris dans un tableau

Tri par bulle

Principe : Le principe consiste à parcourir le tableau en comparant 2 à 2 des éléments voisins et en les permutant éventuellement. Après un balayage complet, le plus grand élément se trouve à l'indice n. On recommence l'opération sur les *n* - 1 éléments restants. On s'arrête lorsque la partie du tableau restant à trier est d'un seul élément ou lorsqu'un balayage complet du tableau n'a pas provoqué de permutation

Par exemple, pour trier le tableau	3 5 2 1 8 9 7 3	
on obtiendra successivement	3 2 1 5 8 7 3 9	5 et 9 "remontent"
	2 1 3 5 7 3 8 9	3 et 8 "remontent"
	1 2 3 5 3 7 8 9	2 et 7 "remontent"
	1 2 3 3 5 7 8 9	5 "remonte"

Ecrire en Scilab une fonction tri_bulle(t)

Le nom 'par bulle" vient du fait que la plus grande valeur remonte petit à petit vers le haut comme une bulle. Cette méthode est efficace pour un tableau presque trié.

Tri par sélection croissant

Principe: Le principe de ce tri consiste à réaliser un premier parcours complet pour rechercher le minimum parmi les *n* éléments d'un tableau, puis un deuxième pour rechercher le minimum parmi les *n* - 1 éléments restants et ainsi de suite. A chaque tour, il faut comparer l'élément de départ, appelé *élément courant, à* tous ses successeurs et effectuer un échange d'emplacement lorsque l'élément courant est supérieur à l'élément utilisé dans la comparaison.

Par exemple, pour trier le tableau	6 2 8 1 5 4	
on obtiendra successivement		
	1 2 8 **6** 5 4	car le minimum est 1
	1 **2** 8 6 5 4	car le minimum est 2
	1 2 **4** 6 5 **8**	car le minimum est 4
	1 2 4 **5** **6** 8	car le minimum est 5
	1 2 4 5 **6** 8	car le minimum est 6
	1 2 4 5 6 **8**	

Ecrire en Scilab la fonction tri-selection(t)

tri par insertion

Principe : étant donné un tableau T, on le trie par ordre croissant en procédant de la manière suivante : chaque élément est inséré dans la liste supposée triée des éléments précédents.

Par exemple, pour trier le tableau	6 2 8 1 5 4	
on obtiendra successivement	6	puis on insère 2
(en n'écrivant que la partie triée du tableau)	2 6	puis on insère 8
	2 6 8	puis on insère 1
	1 2 6 8	puis on insère 5
	1 2 5 6 8	puis on insère 4
	1 2 4 5 6 8	

Description de l'algorithme :

- La variable i désignera l'élément t(i) à insérer dans la liste triée t(1)..t(i-1) (i varie de 2 à N)
- L'élément t(i) sera recopié dans une variable auxiliaire X pour éviter d'être perdu (ou "écrasé") au cours de l'insertion.
- Un deuxième compteur j parcourra l'intervalle [1..i-1] en décroissant pour localiser la place de X (tout élément t(j) supérieur à X sera recopié en t(j+1) puis j sera décrémenté)
- La recherche s'arrêtera dès que l'on aura soit j=0 soit t(j)<=X.
- X sera alors recopié en t(j+1)

Ecrire en Scilab la fonction tri_ins(t)