

ULCO - L2 Mathématiques

DS Algorithmique - 1h00

6 novembre 2018

Exercice 1 : Tri

Le but de cet exercice est de trier un tableau de 100 entiers de telle sorte que :

- les entiers pairs du tableau soient triés dans l'ordre croissant et soient consécutifs sur les indices les plus petits du tableau (= "éléments de gauche");
- les entiers impairs soient triés dans l'ordre décroissant et soient consécutifs sur les indices les plus grands du tableau (= "éléments de droite").

Exemple d'exécution :

```
Tableau de 10 elements non trie : 2 8 7 6 1 0 9 3 5 4
Tableau trie suivant notre methode : 0 2 4 6 8 9 7 5 3 1
```

Autre exemple d'exécution :

```
Tableau de 9 elements non trie : 2 8 2 6 4 0 3 5 4
Tableau trie suivant notre methode : 0 2 2 4 4 6 8 5 3
```

Pour réaliser ce tri, vous allez :

1. sur le modèle du tri rapide, prendre le dernier élément du tableau comme "pivot" et mettre tous les éléments pairs à la "gauche" de ce pivot et tous les éléments impairs à sa "droite";
2. vous testerez si l'élément "pivot" est pair ou impair ce qui permettra d'avoir l'indice "deb_impair" correspondant au premier élément impair du tableau;
3. vous trierez alors les éléments pairs dans l'ordre croissant grâce à un tri par insertion;
4. vous trierez les éléments impairs dans l'ordre décroissant grâce à un tri par fusion.

Bien entendu tous les tris sont à adapter, donc à réécrire entièrement.

Exercice 2 : Les pointeurs

Vous remplirez le tableau ci-après en donnant les valeurs des variables pour chaque instruction du programme mise dans la 1ère colonne.

NOM :

PRENOM :

code	a	b	*p1	*p2	t[0]	t[1]	t[2]	t[3]
int main(void)	?	?	?	?	?	?	?	?
{	?	?	?	?	?	?	?	?
int a, b, *t, *p1, *p2;	?	?	?	?	?	?	?	?
t = new int[4];	?	?	?	?	?	?	?	?
a = 2; b = 3;								
t[0] = 4;								
p1 = t;								
*p1 = a;								
*(t+3) = *t+3;								
p2 = &b;								
*p2 = *(p1++);								
*p1 = a*3;								
*(t+2) = *p2++;								
a = *p1 * *p2;								
return 0;								
}								