

Méthodes de Conception d'Algorithmes

C. TRABELSI & M. FRANÇOIS

TD8 -- Diviser pour Régner : Tri Fusion

Principe du tri fusion

Le principe général du tri fusion est le suivant :

- *on sépare le tableau en deux parties égales (à un élément près dans le cas où le nombre d'éléments n'est pas pair) ;*
- *on trie la première partie ;*
- *on trie la deuxième partie ;*
- *puis on fusionne ces deux parties triées, c'est-à-dire qu'on prend le plus petit élément du premier sous-tableau, le plus petit élément du deuxième sous-tableau et on place le plus petit des deux dans le nouveau tableau et on recommence de la même manière.*

Bien entendu pour trier les deux parties on fait de même, ce qui conduit à un algorithme récursif. Pour qu'il n'y ait pas de problème d'arrêt, il suffit de remarquer qu'un tableau à un seul élément est nécessairement trié.

EXERCICE

- 1. Pour commencer, écrire l'algorithme de la fonction `REPLISSAGE_TAB`. Cette fonction prend en paramètre un tableau `TAB` de taille N puis le remplit avec des entiers aléatoires compris entre 0 et 99 inclus.
- 2. Écrire en C cette fonction dont le prototype est donné par :

```
void REPLISSAGE_TAB (int N, int TAB[N]);
```

- 3. Écrire l'algorithme de la fonction `AFFICHAGE_TAB`. Cette fonction prend en paramètre un tableau `TAB` de taille N puis affiche ses éléments sur la sortie standard.
- 4. Écrire en C cette fonction dont le prototype est donné par :

```
void AFFICHAGE_TAB (int N, int TAB[N]);
```

- 5. Écrire l'algorithme récursif de la fonction `TRI_FUSION`. Cette fonction prend en paramètres un tableau `TAB` de taille N et deux entiers a et b (exp. $a = 0$ et $b = N - 1$) et trie le tableau en utilisant la méthode diviser pour régner.
- 6. Écrire en C cette fonction dont le prototype est donné par :

```
void TRI_FUSION (int N, int TAB[N], int a, int b);
```

- 7. Écrire l'algorithme de la fonction `FUSION`. Cette fonction prend en paramètre un tableau `TAB` de taille N et 3 entiers a, m et b tels que $0 \leq a \leq m < b \leq N - 1$. Les parties du tableau `TAB` entre a et m d'une part, $m + 1$ et b d'autre part étant triées, cette fonction trie la partie résultante du tableau `TAB` entre a et b .

Indications :

- Utiliser un tableau temporaire `TEMP [N]` qui va recevoir au fur et à mesure les éléments triés du tableau original (ici `TAB`).
- Utiliser dans une même boucle deux indices par exemple i et j , pour parcourir les deux parties du tableau en même temps.
- Ne pas oublier de rajouter dans le tableau `TEMP`, la partie du tableau `TAB` qui n'a pas fini d'être explorée.
- Puis pour finir, copier le tableau trié `TEMP` dans le tableau original `TAB`. La copie doit se faire entre les indices a et b inclus.

- 8. Écrire en C cette fonction dont le prototype est donné par :

```
void FUSION (int N, int TAB[N], int a, int m, int b);
```

- 9. Dérouter l'algorithme du `TRI_FUSION` dans le cas où on a :
 $N=4$ et `TAB=[36, 18, 88, 72]`