

Evaluation (1h) - Le drapeau hollandais

On s'intéresse aux méthodes de tri d'un tableau t de taille n fournie par l'utilisateur, contenant seulement 3 sortes d'éléments, notés B (blue), W (white) et R (red). La figure 1 donne un exemple de tableau à trier. La figure 2 indique le résultat attendu. On va étudier 2 approches différentes pour réaliser le travail demandé.

B	B	R	W	W	B	B	R	W	B
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

FIG. 1 – Exemple de tableau de taille $n = 10$ à trier

B	B	B	B	B	W	W	W	R	R
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

FIG. 2 – Résultat à obtenir (les B sont au début du tableau, les R à la fin et les W au milieu)

```
#define B 0
#define W 1
#define R 2

main() { ... }
```

Q1 : Ecrire une fonction `main` dans laquelle on déclare un tableau dynamique de taille n (taille fournie par l'utilisateur `int n; cin>>n;`). On le remplit ensuite avec des valeurs aléatoires. On pourrait considérer que $B = 0$, $W = 1$ et $R = 2$ et travailler directement avec des entiers.

Q2 : Ecrire une fonction `bool seulement_3_valeurs(int *tab, int n)` qui vérifie que le tableau contient bien uniquement des occurrences des trois valeurs B , W et R .

Première méthode

Q3 : Ecrire une fonction `int compteVal(int *tab, int n, int val)` qui compte le nombre d'occurrences de `val` dans le tableau `tab` de taille n .

Q4 : En déduire une fonction `int *produitTableauTrie(int *tab, int n)` qui retourne un *nouveau* tableau où les B , W et R sont triés.

Seconde méthode

La première méthode nécessite de faire 2 itérations sur le tableau (une pour trouver les nombres d'occurrences de chaque élément, et l'autre pour construire le tableau résultat). On va procéder ici avec un simple parcours du tableau de gauche à droite en ré-ordonnant le tableau par des échanges entre positions dans le tableau.

Q5 : Programmer une fonction `void echange(int *tab, int n, int i, int j)` qui échange les contenus de `tab[i]` et de `tab[j]` sous réserve que i et j soient bien dans les bornes du tableau. On testera explicitement ces conditions.

Q6 : On considère 3 variables i , b et r . La variable b marque la fin de la zone (le début du tableau) dont on est sûr qu'elle ne contient que des B . La variable r marque le début de la zone (la fin du tableau) qui ne contient que des R . La valeur i est la variable d'itération permettant de parcourir le tableau case par case. A quelles valeurs faut-il initialiser b , r et i en fonction de la taille n du tableau `tab`? Quelle est la condition d'arrêt de la boucle?

Q7 : Programmer l'algorithme de tri du tableau `void tri_drapeau(int *tab, int n)` à l'aide d'une boucle `while`. On fera bien attention aux bornes des boucles utilisées ainsi qu'aux incrémentations et décrémentations des variables à faire ou non en fonction des conditions d'exécution.