

Feuille de TD n°8 de Programmation 2 (Variantes du tri à bulles)

Exercice1 (TD). Tri à bulles

Question 1. On considère les deux fonctions suivantes.

```
/** Cette fonction échange les valeurs flottantes */  
/* dont les adresses sont reçues en entrées */  
void echange_flottant (float *f1, float *f2) {  
    float tmp = *f1;  
    *f1 = *f2;  
    *f2 = tmp;  
}  
  
/** "Balayage" du tableau de flottants tab de taille n */  
/** Cette fonction modifie le tableau tab */  
void bulle_max (float *tab, int n) {  
    int j;  
    for (j = 0; j < n-1; ++j) {  
        if (tab[j] > tab[j+1]) {  
            echange_flottant(tab+j, tab+j+1);  
        }  
    }  
}
```

a) Quelle est la position de la plus grande des n valeurs de tab après l'appel `bulle_max(tab, n)` ? Calculez, en fonction de n , le nombre minimal et le nombre maximal de comparaisons et d'échanges de valeurs flottantes ? Identifiez les cas correspondant à ces valeurs minimales et maximales.

b) Écrivez la définition d'une fonction

```
void tri_bulle (float *tab, int n)
```

qui reçoit en entrées un tableau de flottants tab et sa taille n et trie les valeurs de tab dans l'ordre croissant en effectuant $n - 1$ balayages de tab , d'amplitudes strictement décroissantes.

c) Calculez, en fonction de n , le nombre de comparaisons de valeurs flottantes requises par l'exécution de `tri_bulle(tab, n)` ? Calculez, en fonction de n , le nombre minimal et le nombre maximal d'appels à la fonction `echange_flottant` ? Identifiez les cas correspondant à ces valeurs minimales et maximales. Dans la suite de l'exercice, l'exécution des instructions de `bulle_max` sera appelée un « balayage » de tab d'amplitude n .

Question 2. On cherche à « améliorer les performances » du tri de la question précédente en diminuant quand c'est possible le nombre de balayages de tab .

a) Supposez que lors d'un balayage de tab exécuté pendant l'appel `tri_bulle(tab, n)`, aucun échange de valeurs flottantes n'ait été effectué. Quelle conclusion peut-on alors tirer sur les valeurs de tab ?

b) Écrivez la définition d'une fonction

```
void tri_bulle_bis (float *tab, int n)
```

qui reçoit en entrées un tableau de flottants *tab* et sa taille *n* et trie les valeurs de *tab* dans l'ordre croissant en supprimant les balayages inutiles de *tab*.

c) Calculez, en fonction de *n*, le nombre de comparaisons de valeurs flottantes requises par l'exécution de `tri_bulle_bis(tab, n)` ? Calculez, en fonction de *n*, le nombre minimal et le nombre maximal d'appels à `echange_flottant` ? Identifiez les cas correspondant à ces valeurs minimales et maximales. Donnez un exemple de tableau sur lequel la fonction `tri_bulle_bis` exécute strictement moins de comparaisons que la fonction `tri_bulle`.

Question 3. On cherche maintenant à diminuer quand c'est possible l'amplitude des balayages de *tab*.

a) Supposez que lors d'un balayage de *tab* exécuté pendant l'appel `tri_bulle_bis(tab, n)`, les dernières valeurs échangées aient occupé les positions *i* et *i* + 1. Quelle conclusion peut-on alors tirer sur les valeurs de *tab* d'indice strictement supérieur à *i* ?

b) Écrivez la définition d'une fonction

```
void tri_bulle_ter (float *tab, int n)
```

qui reçoit en entrées un tableau de flottants *tab* et sa taille *n* et trie les valeurs de *tab* dans l'ordre croissant en diminuant le nombre de balayages *tab* et leur amplitude.

c) Calculez, en fonction de *n*, le nombre de comparaisons de valeurs flottantes requises par l'exécution de `tri_bulle_ter(tab, n)` ? Calculez, en fonction de *n*, le nombre minimal et le nombre maximal d'appels à `echange_flottant` ? Identifiez les cas correspondant à ces valeurs minimales et maximales. Donnez un exemple de tableau sur lequel la fonction `tri_bulle_ter` exécute strictement moins de comparaisons que la fonction `tri_bulle_bis`.

Question 4. Suggérez d'autres améliorations du tri à bulles des questions précédentes.

Exercice 2 (TP)

Reprenez l'exercice de TP de la feuille de TD-TP n°7. Une version complétée de cet exercice est postée sur le site Moodle de l'UE.