CCI - Principes des langages de programmation
TD 6 suite : Tris

François Yvon[*], Thomas Tang[†]

19 octobre 2006

Tris

Exercice 1 *Algorithme de classement des valeurs d'un tableau dans l'ordre croissant par recherche de minimum (tri par extraction)*

1. *Etat initial du tableau, recherche sur l'ensemble du tableau de la plus petite valeur et permutation avec le premier élément.*
2. *Le traitement précédent est de nouveau appliqué mais sur le tableau réduit du premier élément déjà placé.*
3. *Le traitement est identique avec à chaque permutation le tableau réduit d'un élément. le traitment s'arrête lorsque le nombre d'éléments restant à classer est égal à 1.*

Commencez à trier à la main la liste suivant en respectant les règles de l'algorithme : (5,4,7,10,1)
Ecrivez une fonction ***void tri_extraction (int *tab)*** *qui réalise ce tri.*

Exercice 2 *Algorithme de classement des valeurs d'un tableau par permutation et retour en arrière (tri par bulles ou par échange)*

1. *Comparer deux éléments voisins*
2. *Si les éléments sont classés, passage à l'élément suivant (sauf au bas du tableau)*
3. *Si les éléments ne sont pas classés, permutez les deux éléments, et revenez vers l'élément précédent (sauf en haut du tableau où l'on passe à l'élément suivant)*
4. *Le traitement s'arrête quand la comparaison des deux éléments du bas de tableau n'entraîne pas de permutation.*

Commencez à trier à la main la liste suivant en respectant les règles de l'algorithme : (5,4,7,10,1)
Ecrivez une fonction ***void tri_bulles (int *tab)*** *qui réalise ce tri.*

Exercice 3 *Tri rapide ou quicksort*
Voici une implémentation possible en langage C d'un tri rapide.

```
void quickSort(int tableau[], int p, int r) {
        int q;
        if(p<r) {
                q = partitionner(tableau, p, r);
                quickSort(tableau, p, q);
                quickSort(tableau, q+1, r);
        }
}

int partitionner(int tableau[], int p, int r) {
        int pivot = tableau[p], i = p-1, j = r+1;
```

[*] yvon@limsi.fr
[†] ttang@club-internet.fr

```
        int temp;
        while(1) {
                do
                        j--;
                while(tableau[j] > pivot);
                do
                        i++;
                while(tableau[i] < pivot);
                if(i<j) {
                        temp = tableau[i];
                        tableau[i] = tableau[j];
                        tableau[j] = temp;
                }
                else
                        return j;
        }
        return j;
}
```

Quelle est l'idée principale de ce tri ? En connaissez-vous sa complexité ? A partir de cette implémentation, donnez-en une utilisant non pas des tableaux, mais des pointeurs.