Algorithmique Avancée et Langage C

Classe : 3ème Année Ingénieur Génie Informatique

Enseignant : F. HAMDAOUI & S. Ben Ahmed

TP1 ANALYSE DE COMPLEXITE TEMPORELLE

Exercice 1

Soient les trois fonctions ci-dessous.

```
void P1(int n) {
int s = 0;

WHILE (n > 0){
     n = n/3;
     s = s + 1 ;
}
```

```
void P2(int n) {
int s = 0;

FOR(i = 0;i< n; i++) {
    FOR(j= i;j<= n*n;j++){
        s = s + 1;
    }
}
```

```
void P3(int n) {
int s = 0;

FOR(i = 0;i< n;i++) {
  FOR(j= 0;j< i*i; j++){
      FOR(k= 0;k<j;k++){
              s = s + 1;
          }
      }
}
```

Q1. Analyser avec la méthode expérimentale les 3 fonctions chacune dans les trois variantes (pire, meilleure et moyenne des cas) ;

Q2. Refaire le même travail avec la méthode théorique. Comparez les résultats des deux méthodes.

Exercice 2

Q1. Ecrire un programme C qui implémente le tri fusion.

Principe :

Il s'agit à nouveau d'un tri suivant le paradigme diviser pour régner. Le principe du tri fusion (ou tri par interclassement) en est le suivant :

- On divise en deux moitiés la liste à trier (en prenant par exemple, un élément sur deux pour chacune des listes).
- On trie chacune d'entre elles.
- On fusionne les deux moitiés obtenues pour reconstituer la liste triée.

Q2. Ecrire un programme C qui implémente le tri rapide.

Principe :

Le tri rapide - aussi appelé "tri de Hoare" (du nom de son inventeur Tony Hoare) ou "tri par segmentation" ou "tri des bijoutiers" ou, en anglais "quicksort".

- Le principe de ce tri est d'ordonner le tableau en cherchant dans celui-ci une clé pivot autour de laquelle réorganiser ses éléments. Il est souhaitable que le pivot soit aussi proche que possible de la clé relative à l'enregistrement central du vecteur, afin qu'il y ait à peu près autant d'éléments le précédant que le suivant, soit environ la moitié des éléments du tableau. Dans la pratique, on prend souvent le dernier élément du tableau.

- On permute ceux-ci de façon à ce que pour un indice j particulier tous les éléments dont la clé est inférieure à pivot se trouvent dans T(0)...T(j) et tous ceux dont la clé est supérieure se trouvent dans T(j+1)...T(n). On place ensuite le pivot à la position j.
- On applique ensuite le tri récursivement à, sur la partie dont les éléments sont inférieurs au pivot et sur la partie dont les éléments sont supérieurs au pivot.

Q3. Pour chaque programme, donner son ordre de grandeur au pire des cas.
Q4. Pour chaque programme, tester les instances avec taille qui dépasse 10^9.
Q5. Interprétez les résultats trouvés.

Exercice 3

On considère deux manières de représenter ce que l'on appelle des «matrices creuses», c'est-à-dire des matrices d'entiers contenant environ 90% d'éléments nuls :

Q2. a. La matrice est représentée par un tableau à deux dimensions dont les cases contiennent les éléments.

Q2. b. La matrice est représentée par un tableau à une dimension. On ne s'intéresse qu'aux éléments de la matrice qui ne sont pas nuls. Chaque case du tableau contient un triplet (i, j, a) correspondant à l'indice de ligne, l'indice de colonne, et la valeur d'un élément non nul.

Le problème considéré consiste à calculer la somme des éléments d'une matrice.
On demande **d'écrire un programme C** permettant de calculer cette somme, pour chacune des deux représentations, puis de **comparer leur complexité temporelle**.