Centre Universitaire de Mila
Institut des sciences et de la technologie

2eme année L MD informatique
Module A.S.D.1
Année : 2022-

Série de TD N 01
Complexité et récursivité

Exercice 1 : Donner, en fonction du N, le nombre d'affectations, additions, multiplications et de comparaisons effectuées par la fonction puissance suivante:

```
Fonction puissance (a, N : entier) : entier
  i, R : entier.
Début
    i← 0 ; R← 1;
    Tantque i < N faire
        R ← R * A; i←i+1 ;
    Fintantque
    Retourne (R) ;
Fin
```

Exercice 2 : Quel est, en fonction du N, le nombre d'affectations (R ← R + 1) effectuées par l'algorithme sous-dessous? Que calcule cet algorithme. Donner, si possible deux algorithmes plus rapides que cet algorithme.

```
Algorithme Exemple
I, J, N, R : entier
Début
    R← 0 ;
    Pour I de 1 à N faire
        Pour J de 1 à I faire
              R ← R + 1 ;
        Finpour
    Finpour
    Ecrire (R)
Fin
```

Exercice 3 : Que fait le programme suivant ? Évaluez sa complexité.

```
int main ( )
{       float prix, quantite, montant ;
        cin>> prix>>quantite;
        montant = prix*quantite ;
        if (quantite <= 350)
            cout << montant;
        else
             if (quantite > 350 && quantite <=600 )
             {
                  montant=montant-montant*0.02 ;
                  cout << montant;
             }
             else
                 cout << montant- montant*0.03;
}
```

Exercice 4 : écrire la fonction qui retourne la valeur maximale d'une matrice d'entiers n×m. Combien de temps prend cette fonction pour trouver la valeur minimale d'une matrice de $10^3 * 10^3$ éléments. On considère que la vitesse de la machine utilisée est 10^6 opérations par seconde.

Exercice 5 : donner les classes de complexité des fonctions suivantes

1. $F(n) = 3n + 2\log(n) + 10$
2. $T(n) = 3n^2 + 5$
3. $C(n) = n \log(n) + 8n + 3$
4. $P(n) = 2^n + 10\,n^3 + 8$
5. $Q(n) = 15$

Exercice 6 : évaluer la complexité dans le pire de cas pour la fonction recherche sous-dessous, qui permette de rechercher un élément dans un tableau trié. Comparer cette complexité avec la complexité d'une fonction basée sur une recherche séquentielle.

```
Fonction recherche_dichotomique(x :entier, N : entier, T[] : tableau d'entiers) : booléen
    inf, sup, m: entier     b : booléen
Debut
    inf ←1;      sup ← N;    b←faux ;
   Tant que (inf <= sup et b=faux) faire
         m ← (inf + sup ) / 2;
         Si (elt = T[m]) alors
            B←vrai ;
         Sinon
              Si  (elt < T[m]) alors
                   sup ← m - 1;
              Sinon
                  inf ← m + 1;
              Fin Si
         Fin si
    FinTque
         Retourne b ;
Fin
```

Exercice 7 : évaluer la complexité de la fonction récursive suivante.

```
int puissance (int x, int n)
{
    if (n==0)
         return 0;
    if (n==1)
         return x;
    else
         return x + puissance (x, n-1);
}
```

Exercice 8 : écrire permettant de saisir un tableau d'entiers et ensuite afficher la somme de ses éléments. Évaluer sa complexité.