**Université IBN KHALDOUN –TIARET- Faculté Des Mathématiques et de l’informatique**

**Département d’informatique, 2ière Année Licence : -2014/2015-**

## TD N°01 : Analyse de la complexité

**Exercice 1 : Fouille exhaustive**

Il existe des outils qui peuvent “essayer” approximativement 500.000 mots de passe à la seconde.

1. Calculer le nombre de possibilité et temps d’exécution pour les tailles de données suivantes :

4 chiffres, 6 chiffres, 8 chiffres et 16 chiffres.

1. Ecrire un algorithme qui explore toutes les combinaisons possibles codées sur deux chiffres
2. Ecrire un algorithme qui explore toutes les combinaisons possibles codées sur deux chiffres

**Exercice 2 :** Déterminer la complexité de chaque algorithme :

1. fonction de recherche d’un élément dans un tableau non ordonné (recherche séquentielle)
2. fonction de recherche d’un élément dans un tableau ordonné (recherche dichotomique)
3. fonction qui calcule le produit de deux matrices carrées A et B de n lignes et n colonnes

# Exercice 3 : La fonction peut être approchée au voisinage de par son développement limité

# On considère la variante suivante :

Fonction exponentielle(x : Réel ; n :Entier) :reel

# Var ex, p :réel ;

i, j :entier ;

Début

Ex 🡨 1 ;

##### Pour i ≤ 1 à n faire

P 🡨 1 ;

Pour j ≤ 1 à i faire

P 🡨 p.x/j

Finpour

##### Finpour

Retourner(ex + p);

Fin

1. Montrer que la complexité de la fonction ainsi définie est.
2. Améliorez cette fonction de telle façon à réduire au maximum sa complexité (on peut facilement obtenir un calcul de complexité linéaire).

**Exercice 4**

1. Donner la complexité (en notation Grand O) pour chacun des quatre fragments de programme suivants. A noter que le temps d’exécution correspond ici au nombre de fois où l'opération sum ++ est exécutée. sqrt est la fonction qui renvoie la racine carrée d'un nombre donné.

|  |  |  |  |
| --- | --- | --- | --- |
| (a) sum = 0; for(i=0;i<sqrt(n)/;i++)  sum++;  for(j=0;j<sqrt(n)/4;j++)  sum++; for(k=0;k<8+j;k++) sum++; | (b) sum = 0; for(i=0;i<sqrt(n)/2;i++)  for(j=i;8+i;j++)  for(k=j;k<8+j;k++) sum++; | (c) sum = 0; for(i=1;i<2\*n;i++)  for(j=1;j<i\*i;j++) for(k=1;k<j;k++)  if (j % i == 1)  sum++; | (d) sum = 0; for(i=1;i<2\*n;i++) for(j=1;j<i\*i;j++) for(k=1;k<j;k++) if (j % i)  sum++; |

**Exercice 5:** Déterminer de manière intuitivelacomplexité des deux problèmes suivants :

**Problème SAC à DOS**: M (limite de poids)=14, Solution optimale > P1,P2,P3,P5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **P1** | **P2** | **P3** | **P4** | **P5** | **P6** | **P7** | **P8** |
| Valeur | 10 | 5 | 1 | 9 | 3 | 4 | 11 | 17 |
| Poids | 7 | 3 | 3 | 10 | 1 | 9 | 22 | 15 |

**Problème partitions :** Entrées : ensemble des enties positifs S, Sortie: S1 et S2 telque S1∩S2=∅, S1∪S2=S, S= {1, 7, 10, 9, 5, 8, 3, 13} Solution optimale : S1= {1, 10, 9, 8} S2={7, 5, 3, 13}