Université d'Auvergne - IUT Clermont Ferrand

Département Informatique

TD4 : Complexité temporelle

**Rappel :**

**Liste des opérations élémentaires :**

* Les opérations arithmétiques (+, -, \*, /)
* Les affectations (a = 2;)
* Les tests logiques (==, <, >, …)

**Ordre de grandeur :**

* : Constante
* : Logarithmique
* : Linéaire
* : Polynomiale (k constante)
* : Exponentiel (k constante)
* : Factoriel (TSP, VRP)

**Trouver l’ordre de grandeur :**

for(int j= 1; j <= n; j++){

for(int i = 1; i <= n; i++){

int a = i + j ;

a = a \* 3 ; 5 opérations ne dépends pas de n O(n) O(n²)

System.out.println(a); --> O(1)

}

}

**Pour les exemples suivants, donner leur algorithme et leur complexité temporelle en ordre de grandeur.**

1. **Test de primalité**

Le but : Tester si un entier est un nombre premier.

Principe : Tester pour chaque valeur plus petite que sa racine, si elle est un dénominateur valide

1. **La recherche par dichotomie**

Le but : la recherche d’un élément dans un tableau d’entier **triés**.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 14 | 34 | 45 | 48 | 54 | 63 | 86 | 94 |

Principe pour trouver un élément :

1. On regarde celui du "milieu" du tableau à chaque étape :
   1. On s’arrête si c’est le bon
   2. On explore la moitié inférieure si l’élément cherché est plus petit que celui au milieu
   3. On explore la moitié supérieur si l’élément cherché est plus grand que celui au milieu

Exemple chercher 14 :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 14 | 34 | 45 | 48 | 54 | 63 | 86 | 94 |

14 < 45

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 14 | 34 | 45 | 48 | 54 | 63 | 86 | 94 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 14 | 34 |

14 > 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 14 | 34 |

|  |  |
| --- | --- |
| 14 | 34 |

14 trouvé

1. **Enumération des solutions pour TSP**

Une solution est une liste ordonnée de tous les clients :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 3 | 4 | 7 | 2 | 9 | 10 | 5 | 11 | 8 | 6 |

Principe : L’ensemble des solutions correspond à l’ensemble des permutations existantes pour la liste des clients. Il suffit donc de toutes les listées.