

Centre Universitaire de Mila
Institut de mathématiques et informatique
Département de l'informatique

Master 1 I2A

Année : 2023/2024

Matière : Algorithmique avancée et complexité

TD 1

Exercice 1 :

Donner l'ordre de grandeur de la complexité des trois algorithmes suivants :

Algorithme 1.

```
S ← 0;
Pour i = 1 à n faire
    S ← S + i;
FinPour
retourner S;
```

Algorithme 2.

```
S ← 0;
Pour i = 1 à n faire
    Pour j = i à 10 faire
        S ← S + j;
    FinPour
FinPour
retourner S;
```

Algorithme 3.

Entrées: T : matrice d'entiers trié par ligne ; n, S : entier;

Sortie: booléen;

i ← 1; trouver ← faux;

Tant que (i ≤ n Et non trouver) faire

Min ← 0; Max ← T[i];

Tant que (non trouver Et Min < Max) faire

Avg ← partieEntière ((Min + Max) / 2);

Si (T[i][Avg] = S) alors trouver ← vrai;

Sinon

Si (T [i][Avg] < S) alors Min ← Avg + 1;

Sinon Max ← Avg - 1;

FinSi

FinSi

FinTant que

i ← i + 1;

FinTantque

Retourner trouver ;

Exercice 2 :

1. Donner l'ordre de grandeur des fonction suivantes :

1) $f(n) = n^4 - n + 1$;

2) $f(n) = 1000n^4 + 2 \times n!$

3) $f(n) = n^2 + n \times 2024$;

4) $f(n) = 15! \times n + 4 \times n^3$;

5) $f(x) = n^9 + 3^{2n} + 2023 + 2 \times n + 5 \times \log(n)$;

2. Dire si chacune des affirmations suivantes est vraie ou fausse :

A1 : $n \in O(n \log(n))$?

A2 : $4n^{2n} \in O(n^3)$?

A3 : $n / \log(n) \in O(n)$?

A4 : $100n^2 + n + 5 \in O(n^2)$?

A5 : $4^n + 3^n \times 6 \in O(5^n)$?

Exercice 3 :

On veut implémenter une table de hachage (de taille n) pour stocker des nombres entiers qui sont inférieurs à $n \times 2$ et divisibles sur 3. En considérant l'adressage ouvert avec un sondage linéaire pour gérer les collisions :

- 1) Proposer une fonction de hachage ;
- 2) Ecrire l'algorithme qui implémente l'insertion d'un entier dans cette table ;
- 3) Ecrire l'algorithme qui cherche si un entier existe dans cette table ou non ;
- 4) Ecrire l'algorithme qui permet de supprimer un entier de cette table.