

TD2

Data Structure & Algorithm I



Matthieu Jimenez

Été 2015

TD2

Algorithm I

Exercice I

Calculer:

$$15n+12=O(n)$$

$$17n^2+3n+4=O(n^2)$$

$$25n^3+20\log(n)=O(n^3)$$

Exercice II

Enoncé:

Calculer:

$$2n^3+10n+17=\Theta(n^3)$$

$$n/3+15=\Theta(n)$$

Exercice III

Enoncé:

Une méthode simple pour découvrir le mot de passe d'un compte est d'utiliser le Brute Force, c'est à dire de tester toutes les possibilités de mots de passe existants. En supposant que le mot de passe que vous tentez de découvrir par Brute Force est composé de n caractères alphanumériques (A-Z,a-z,0-9), quel sera la complexité dans le pire des cas du code que vous utiliserez?

Exercice IV

Énoncé:

On veut écrire un algorithme qui décide si un mot est un palindrome ou non. Un mot est un palindrome si sa première lettre est identique à la dernière, sa deuxième à l'avant dernière etc...

1. Proposez un pseudo-code
 - input: Chaîne de caractères
 - output: boolean

2. Quelle est la complexité de votre algorithme?

Exercice V

Énoncé:

Quel sera la complexité de cet algorithme?

Donner directement sous la forme Big O

```
int sum = 0;
for (int i = 1; i <= N; i++)
    for (int j = 1; j <= i*i; j++)
        for (int k = 1; k <= j; k++)
            sum++;
```