



L3 LFTI - Exercices d'algorithmique : premières notions de complexité

Karën Fort

31 janvier 2017



Important : vous devrez rendre à la fin de cette séance les algorithmes demandés. Vous devez chez vous en écrire la version Java et rendre un fichier nommé `TD2_NomPrenom.java` contenant les méthodes demandées dans les exercices ainsi qu'une méthode *main* qui permettra d'exécuter votre code sur un exemple. Vous pouvez utiliser Eclipse (<https://www.eclipse.org/>) ou NetBeans (<https://netbeans.org/>) ou n'importe quel logiciel de votre préférence, mais la responsable de TD ne l'installera et ne le configurera pas pour vous.

Attention : tout plagiat (notamment, sur Internet) ou copie directe sur le voisin ou la voisine sera sanctionné par un 0. Les communications entre vous sont autorisées, mais le code doit être le vôtre.

1 Maximum dans un tableau

Étant donné un tableau d'entiers (par exemple, celui qui vous est proposé) :

1. Trouvez l'algorithme le moins complexe possible permettant de trouver l'entier le plus grand
2. Évaluez sa complexité

```
public class RechercheMaximum {  
  
    public static void main(String[] args) {  
        int tabEnt[] = {76, 12, 10, 21, 65, 45, 6, 54, 87, 17};  
        // ici votre code  
    }  
}
```

2 Anagrammes

Une anagramme est un mot obtenu par permutation des lettres d'un mot. Par exemple le mot « elisa » possède comme anagrammes « ailes », « asile », « esial », « salies »,.... Bien sûr, parmi ces anagrammes certains mots n'ont aucun sens en français, comme par exemple aeils. Étant donné un mot, on appelle anagramme de ce mot, tout autre mot obtenu en plaçant les lettres dans un ordre quelconque. On ne s'occupe pas de savoir si le mot obtenu est sensé ou non.

1. Combien y a-t-il d'anagrammes pour le mot « MANOIR » ?
2. Écrivez un algorithme qui rend vrai si la chaîne de caractères s1 est une anagramme de la chaîne de caractères s2, faux sinon (le squelette du code est donné ci-dessous).
3. Évaluez sa complexité

Bonus : si votre algorithme a une complexité supérieure, cherchez un algorithme en $O(n)$

```
public class Anagramme {

    public static void main(String[] args) {
        String s1 = "parisien", s2 = "aspirine";
        \\ici votre code
    }
}
```

3 Bonus : palindromes

On appelle « palindrome » un mot ou une phrase qui se lit de la même façon à l'endroit comme à l'envers, sans tenir compte des espaces. Exemple : le mot « ABCBA » est un palindrome. La phrase « ESOPE RESTE ET SE REPOSE » (sans tenir compte des espaces on obtient le mot « ESOPE-RESTEETSEREPOSE ») se lit de façon identique de la gauche vers la droite ou de la droite vers la gauche.

1. Écrivez une fonction itérative qui rend vrai si la chaîne de caractères est un palindrome, faux sinon
2. Écrivez une forme récursive correspondant à la forme itérative précédente