

# Moteurs de recherche

## TP1

Josian CHEVALIER  
Thomas SALMON

16 février 2015

### 1 Exercice 1

Le type matrice peut être initialisé de deux façons :

- En passant en paramètre de son constructeur les tableaux C, L et I. Dans ce cas elle sera directement initialisée avec les bonnes valeurs mais cela implique de calculer C, L et I au préalable
- En passant en paramètre de son constructeur la taille de la matrice, auquel cas toutes les cases seront initialisées à 0. On pourra modifier leurs valeurs à l'aide de la fonction `changeValue()`, afin de modifier les valeurs case par case.

L'initialisation d'une matrice est en temps constant. La récupération ou la modification d'une valeur est en  $O(n)$ , car on parcourt au maximum une sous section de I, qui représente une ligne de la matrice.

La multiplication par un vecteur se fait à l'aide de la fonction `mult_vect()`. Elle parcourt L (de taille n), et pour chaque ligne parcourt la section de C et I qui correspond à la ligne. Ainsi on parcourt Une fois L, et une seule fois chaque valeur de I (et de C). La complexité de cette fonction est donc de  $O(n+m)$ .

### 2 Exercice 2

Ce calcul s'effectue au travers de la fonction `mult_vect_transp()`. L'algorithme est le même que pour le calcul de la multiplication par vecteur, on change simplement les éléments du vecteurs résultats affectés à chaque boucle. Ainsi, on garde une complexité en  $O(n+m)$ .

### 3 Exercice 3

Pour faire un algorithme calculant la probabilité d'être sur les différents sommets en un certain nombre de pas, nous avons légèrement modifié PageRank. Ainsi

sa condition d'arrêt n'est plus l'atteinte de la marge d'erreur acceptable, mais on boucle simplement autant de fois que l'on souhaite avoir d'étapes.

La classe PageRank contient les méthodes permettant d'effectuer ces traitements :

- La méthode 'rank()' permet de partir de la liste des probabilité de se trouver sur chaque sommet, et calcule pour un nombre de pas les probabilités en fonction d'une matrice.
- La méthode 'rank\_from\_sommet()' utilise la méthode précédentes en lui fournissant une liste ou la probabilité d'être sur chaque sommet est de 0, sauf pour le sommet 'num\_sommet', pour laquelle elle est de 1.
- La méthode 'rank\_zero()' utilise la méthode précédente en lui indiquant que l'on part du sommet 0.

## 4 Exercice 4