Choix d’implémentations

Séparation des sommets en deux

Le graphe a été conçu pour les Sommets qui sont ou des Utilisateur, ou des Page. Ce choix nous donne une structure interne à 2 ensembles, un pour les Utilisateur et l’autre pour les Page.

Ce choix nous donne une amélioration quand on demande la liste des utilisateur, ou la liste des pages. Certaines autres fonctions, comme la moyenne d’âge, ne regarde pas dans les mauvais ensembles inutilement.

L’interface graphique est aussi conçu pour refléter ce choix.

Implémentations des arcs double

Les arcs sont implémenté doublement : les sommets se souviennent de leur voisins sortant, et les arcs sont implémenté par une classe interne non mutable du graphe, dont on met les instances dans un ensemble.

Cette implémentation est malpropre, et une réécriture pour que le graphe gère lui même les arcs plus proprement est envisageable.

Une liste d’adjacence sous forme de Map<Sommet, LinkedList<Sommet>> est aussi disponible, bien que redondante, mais elle ne s’initialise pas automatiquement.

Unicité des noms

On exige que chaque sommet possède un nom unique dans le graphe. Cette propriété nous permet de trouver aucun ou au plus un sommet lorsque on cherche si un nom appartient a un sommet du graphe. Cela évite l’ambiguïté.

Algorithmes Principaux

PageRank

Cet algorithme a été implémenté comme demandé par une classe sans constructeurs accessible, qui renvoi une Map qui associe chaque sommet à son PageRank calculé.

Comme on ne peut pas accéder au voisins entrant d’un sommet directement, on initialise une liste des prédécesseurs. Cela coûte en complexité, et une piste d’amélioration serait de changer les donnée pour accommoder une demande des prédécesseurs plus facilement.

Avec N = nombres de sommet, et M = nombres d’arcs, la complexité de la création de la liste des prédécesseurs est de l’ordre de N^2. Celui du PageRank à proprement parler est de l’ordre (N + M).

Sauvegarde du graphe en fichier

DÉCRIRE ICI LA SAUVEGARDE EN FICHIER //OK

Moyenne d'âge des utilisateurs

Si on a au moins un utilisateurs, on peut afficher une moyenne d'âge. On fait tout simplement (La somme d'âge des utilisateurs) / (nombre d'utilisateur).

Cette formule simple pourrait être amélioré pour éviter le débordement.

Trouver un sommet par nom

Pour trouver un sommet par son nom, on cherche tout simplement par tout les sommet si l'un d'eux à un nom égal à la recherche. Complexité : linéaire avec les nombres de sommets.

Distance minimale à partir de s

L'algorithme est implanté comme suit : un sommet s possède une Map qui lui indique la distance à un sommet u. Cette Map peut être rafraîchi depuis le graphe en utilisant computeSmallestDistance(s).

La complexité de computeSmallestDistance est (avec N nombre de sommet et M nombre d'arc) : (2*N) + M.

Difficultés rencontrés

Coordination

Ce projet étant le premier projet collaboratif pour nous, des problèmes de coordination au niveaux des décisions, des changements au code, et de la répartition du travail ont été présents.

Affichage du graphe

L’affichage graphique étant assez compliqué, l’utilisation d’une librairie externe a été considéré.

// Elle n’a pas abouti cependant. // Malheureusement