TM : Algorithme PageRank

Le PageRank ou PR est l'algorithme d'analyse des liens concourant au système de classement des pages Web utilisé par le moteur de recherche Google. Il mesure quantitativement la popularité d'une page web. Le PageRank n'est qu'un indicateur parmi d'autres dans l'algorithme qui permet de classer les pages du Web dans les résultats de recherche de Google. Ce système a été inventé par Larry Page, cofondateur de Google. Ce mot est une marque déposée.

Source : wikipédia

1 Simulation avec des `Spiders`

le principe de base est d'attribuer à chaque page une valeur proportionnelle au nombre de fois que passerait par cette page un utilisateur parcourant le Web en cliquant aléatoirement sur un des liens apparaissant sur chaque page.

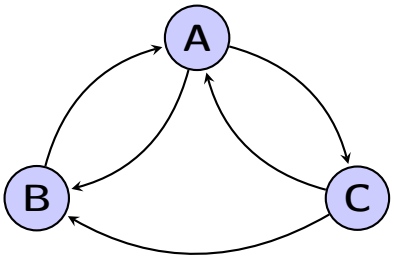
Nous allons simuler les clics d'un tel utilisateur en utilisant des robots, appelés *spiders* qui se déplaceront sur des pages d'un mini-web créé.

On dispose d'une classe `Page` permettant de représenter une page internet. Elle dispose d'un champ `pagerank` initialisé à `0`, d'un champ `url` correspondant à sont url ainsi qu'un tableau `pagesVoisines` des pages internet accessibles par des liens présents sur la page considérée.

```
class Page{
    int pagerank;
    String url;
    Page[] pagesVoisines;
}
```

1. Définir une classe `Web` permettant de représenter un ensemble de Pages. Elle dispose de deux champs `nombrePages` (le nombre de pages web de la toile) et `pages` (les pages qui composent la toile).

2. Définir le `web` ci-dessous représentant les liens entre trois pages A, B et C.



3. Définir la classe `Spider` représentant une araignée se déplaçant sur la toile. La structure dispose des champs suivants :

   — `page` : la page internet sur laquelle elle se trouve

   — `nbPageVisitees` : le nombre de pages à visiter.

   `pagesVisitees` : le tableau des pages visitées.

4. Dans la classe `Web`, écrire une méthode `void PageRank(int n)` qui estime le PageRank de chaque page du web en effectuant $n-1$ déplacements d'un `spider` et en attribuant la fréquence de présence du spider sur la page au champ `pagerank` de chaque page.
   L'araignée est placée sur une page aléatoire et va visiter n pages au total A chaque étape, l'araignée se déplace sur une des pages accessible (depuis la page actuelle) de manière équiprobable. Si aucun lien n'est

disponible, elle se place à l'étape suivante sur une page aléatoire.

5. Ecrire une méthode `String toString()` qui "affiche" les url des pages web ainsi que leur PageRank, par ordre décroissant de PageRank.

6. Déterminer alors une estimation du Pagerank du `miniweb` défini précédemment pour différentes valeurs de n. Que remarquez-vous ? Cela vous semble-t-il cohérent ?

7. Un utilisateur ne suit les liens de page en page que 20 fois avant de repartir sur une nouvelle page quelconque. Modifier le programme pour tenir compte de cette remarque.

8. optionnel : pour mieux tester vos fonctions, il faudrait un web plus important. Vous aller ajouter à la la classe `Web` une méthode qui ajoute une nouvelle page voisine. Attention, qu'est-ce qu'on fait quand le tableau est plein ? En utilisant cette méthode écrire une fonction qui prend en paramètre la taille de votre web, crée un tableau de pages web, ensuite pour chaque page, elle ajoute un nombre aléatoire de voisins, choisis de façon aléatoire dans le tableau.

2 La formule du PageRank

Voici quelques idées générales sur ce que doit refléter le PageRank d'une page :

— L'idée principale est que si une page A fait un lien vers une page B, alors c'est que la page A juge que la page B est suffisamment importante pour mériter d'être citée. Autrement dit, un lien de la page A vers la page B doit augmenter le PageRank de la page B.

— De plus, l'augmentation du PageRank de la page B doit être d'autant plus importante que le PageRank de la page A est élevé.

— Enfin, l'augmentation du PageRank de la page B est d'autant plus importante que la page A fait peu de liens. En d'autres termes, si la page A juge que peu de pages méritent un lien et que la page B fait partie de ces liens, alors il est normal que le PageRank de la page B augmente plus que dans le cas où de nombreuses pages obtiennent un lien.

Intéressons nous maintenant à la formulation mathématique de ces idées.

Remarque : l'algorithme du PageRank a beaucoup évolué depuis l'article rédigé par les deux fondateurs de Google.

Soient A_1, A_2, ..., A_n n pages pointant vers une page B et :

$PR(A_k)$ le PageRank de la page A_k,
$N(A_k)$ le nombre de liens sortants présents sur la page A_k,
d un facteur compris entre 0 et 1, fixé en général à 0,85.

Alors le PageRank de la page B se calcule à partir du PageRank de toutes les pages A_k de la manière suivante :

$$PR(B) = (1 - d) + d \times \sum_{k=1}^{n} \frac{PR(A_k)}{N(A_k)}$$

Comment utiliser cette formule pour calculer d'une autre manière le PageRank des pages du `web` ?

Remarque : on pourra commencer par initialiser le PageRank de chaque page à 1 avant d'appliquer la formule précédente.