

TP - Représentation des données-Les dictionnaires

Nom :

Le Rang d'une page Web...

En 1998, deux jeunes doctorants de l'université de Stanford, Larry Page et Sergey Brin (en collaboration avec Motwani et Wynograd) publiaient un article intitulé "The PageRank Citation Ranking: Bringing Order to the Web" présentant les résultats d'un nouvel algorithme permettant de classer les pages web selon leur popularité et montrant la précision de cet algorithme sur un nouveau moteur de recherche appelé ... Google !

Voilà une version simplifiée de cet algorithme:

- On simule le comportement d'un internaute
- On démarre au hasard sur une page
- On suit de façon aléatoire un lien sur cette page
- Et ainsi de suite...

On impose le fait que dans 15% des cas , l'utilisateur abandonne sa navigation pour repartir d'une page au hasard.

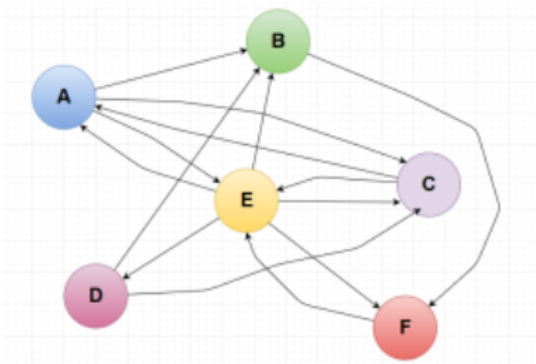
Notre utilisateur va se balader de pages en pages, en faisant son petit tour du net. A chaque fois que l'utilisateur tombe sur une page donnée, cette page gagne un point. Et à la fin, la page ayant le plus de point est alors la page la plus populaire du réseau !

Un cas en particulier

Considérons 6 sites web : A, B, C, D, E, F ayant des liens hypertexte entre eux.

- Le site A contient un lien vers les sites "B", "C" et "E" (3 liens sortants)
- Le site B contient un lien vers le site "F" (1 lien sortant)
- Le site C contient un lien vers les sites "A" et "E" (2 liens sortants)
- Le site D contient un lien vers les sites "B" et "C" (2 liens sortants)
- Le site E contient un lien vers les sites "A", "B", "C", "D" et "F" (5 liens sortants)
- Le site F contient un lien vers le site "E" (1 lien sortant)

Cela peut se modéliser à l'aide d'un graphe :



? QUESTION 1:

D'après vous quel est le site le plus populaire?

Classer à priori les sites par ordre de popularité.

Implémentation des données

Nous commencerons par stocker les noms de sites dans une liste

```
#liste des sites web
```

```
Websites = ["A", "B", "C", "D", "E", "F"]
```

Les dictionnaires sont particulièrement adaptés pour représenter les sites et leur liens

À FAIRE 1:

Créer un dictionnaire Hypertext dont les clés sont les sites web et les valeurs la liste des sites vers lesquels ils ont un lien sortant
exemple : `Hypertext["A"] = ["B", "C", "E"]`

À FAIRE 2:

Créer un dictionnaire Walk_number dont les clés sont les sites web et les valeurs, initialisées à 0, sont le nombre de fois où le site a été visité
exemple : `Walk_number["A"] = 0.0`

Le calcul...

Voici l'algorithme de calcul:

```
import random
i=0
tant que i<1000
    On choisit au hasard un site -> x
    Tant qu'un nombre aléatoire est inférieur à 0.85
        Le nombre de visite de x est incrémenté de 1
        x <- un lien hypertexte visitable choisi de façon aléatoire
    i=i+1
On fait afficher les résultats
```

- L'instruction `random.random()` renvoie un nombre dans l'intervalle $[0,1[$
- L'instruction `x = random.choice(Websites)` permet de choisir au hasard un site et le stocke dans x
- L'instruction `x = random.choice(Hypertext[x])` permet de

 **À FAIRE 3:**

Traduire cet algorithme en Python et donner le classement des sites web obtenus

Écrire votre programme ici