IV - Les moteurs de recherche

a) Présentation

Une recherche sur le web se fait à l'aide d'un moteur de recherche. Il ne faut pas confondre le navigateur et le moteur de recherche :

- un navigateur permet d'afficher du contenu web après avoir effectué une requête auprès d'un serveur ;
- le moteur de recherche utilise le navigateur pour effectuer des recherches.

Cette vidéo explique de manière un peu décalée l'historique des moteurs de recherche et leur mode de fonctionnement pour obtenir des résultats pertinents.
https://www.youtube.com/watch?time_continue=3&v=0A5fQER40Wg

N'oubliez pas qu'il n'existe pas un unique moteur de recherche, même si en France Google est utilisé dans plus de 90 % des cas.
Sachez cependant que les résultats renvoyés lors de requêtes peuvent être différents d'un moteur de recherche à l'autre sans pour autant pouvoir dire que l'un est meilleur qu'un autre. Yahoo et Bing sont aussi connus, même si leur utilisation est très minoritaire par rapport à Google.
Il en existe par ailleurs d'autres dont la philosophie est très différentes.

Travail : Faire des recherches sur d'autres moteurs de recherches et donner leurs avantages (et éventuellement quelques inconvénients) par rapport à ceux dont nous avons parlé précédemment (répondre sur Jupyter) :

- DuckDuckGo ;
- Qwant.

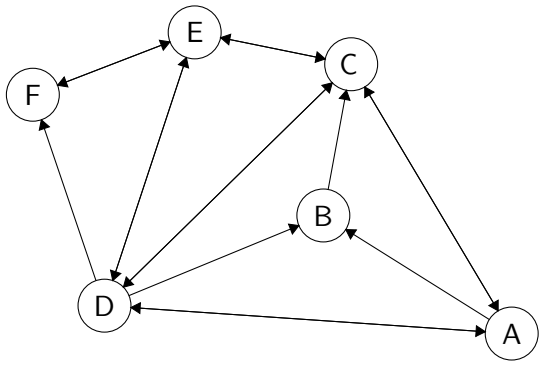
Pour vous centraliser le travail que vous avez à rendre et parce que dans la suite, je vais vous demander de programmer en python, vous aller utiliser l'application Jupyter présente dans lyceeconnecte (une vidéo de présentation est proposée sur mon site).

b) Principe du parcours du Web par un moteur de recherche

Le Web est constitué de pages reliées entre elles par des liens hypertextes. Cela peut être représenté par un graphe où chaque sommet serait une page. Les moteurs de recherche parcourent constamment ce graphe (c'est-à-dire toutes les pages du Web) afin de collecter un maximum de contenu qu'ils indexent. C'est ainsi qu'il peut fournir une réponse rapide et pertinente.

En 1998, les informaticiens Larry Page et Sergey Brin proposent **l'algorithme *PageRank*** qui a conduit à la création du moteur de recherche Google. La popularité d'une page est obtenue à partir de la popularité des pages qui la citent.

Exercice : On a représenté ci-contre les liens présents sur 6 pages Web sous la forme d'un graphe orienté. La flèche qui va de D vers F signifie que la page D contient un lien vers la page F.
Un internaute arrive au hasard sur l'une de ces pages. Il suit de manière aléatoire les liens proposés par la page augmentant ainsi le nombre de vues. Dans 80 % des cas, il continue la navigation (le reste du temps, il ferme son navigateur). On s'intéresse à la page ayant la plus grande popularité suite à la navigation de 1000 personnes.



Cette simulation sera réalisée en python.
Retourner dans le fichier Jupyter dont il a été question dans la partie précédente. Dans ce programme :

- « Lien » permet d'identifier les flèches du graphes (donc les pages accessibles par un lien depuis une autre page). Par exemple `Lien["A"]=["B", "C", "D"]` car de la page A, on trouve 3 liens qui permettent d'accéder aux pages B, C et D.
- La première boucle `for` permet d'initialiser le nombre de clics de chaque page à 0. C'est équivalent à `NbClic[ "A" ]=0, NbClic[ "B" ]=0, ....`

– La deuxième boucle `for` permet de simuler le comportement des 1000 internautes. Chacun d'eux va sur une de ces 6 pages et dans 80 % des cas, il poursuit sa navigation après un changement de page (d'où la boucle `while`).

– La dernière boucle est juste là pour afficher les résultats sous la forme de phrases (une par page).

```
1  from random import random, choice
2  Pages=["A", "B", "C", "D", "E", "F"]
3  Lien={}
4  Lien["A"]=["B", "C", "D"]
5  Lien["B"]=[  ]
6  Lien["C"]=[  ]
7  Lien["D"]=[  ]
8  Lien["E"]=[  ]
9  Lien["F"]=[  ]
10 NbClic={}
11 for page in Pages:
12     NbClic[page]=0
13 for i in range(1000):
14     choix=choice(Pages)
15     NbClic[choix]=NbClic[choix]+1 #on ajoute 1 au compteur de la page choisie au hasard
16     while random()<0.8:
17         # cela traduit le fait que dans 80 % des cas l'internaute poursuit sa navigation.
18         # Donc on fait un deuxième choix aléatoire, mais seulement parmi les liens pré-
   sents sur la page choisie
19         choix=choice(Lien[choix])
20         NbClic[choix]=NbClic[choix]+1
21 #La simulation des 1000 internautes ayant été faite, on affiche les résultats
22 for page in Pages:
23     print("La page", page, "a été visitée", NbClic[page], "fois.")
```

Répondre aux questions suivantes (sur le fichier Jupyter) :

1. La première ligne de ce programme permet de charger deux fonctions du module random : random et choice. Que fait chacune de ces deux fonctions ?

2. Compléter les lignes 5 à 9 indiquant les pages accessibles depuis les pages B à F. Regarder ce qui est fait pour la page A (compiler la ligne pour vérifier que cela fonctionne).

3. Décrire sous la forme d'une phrase ce à quoi correspond `NbClic["A"]`.

4. On s'intéresse au pourcentage de clics sur chacune des pages.
   Ce pourcentage étant égale à `NbClic[x]/TotalClic` où `TotalClic` devrait être déterminé dans le programme (`x` prenant les différentes valeurs `"A"`, `"B"`, ...).

   (a) Deux lignes devront être ajoutées dans ce programme :
   `TotalClic=0`
   `TotalClic=TotalClic+1`
   Recopier le programme dans une nouvelle ligne et ajouter chacune de ces deux lignes au bon endroit à chaque fois (la deuxième devant être ajoutée deux fois).

   (b) Dans la dernière boucle du programme, faire afficher le pourcentage de popularité de chaque page.

   (c) Quel est la page la plus populaire et quel est environ son pourcentage de popularité (vous pourrez augmenter le nombre d'internautes et vérifier que le résultat obtenu n'est pas très éloigné de ce qui est obtenu avec 1000 internautes) ?