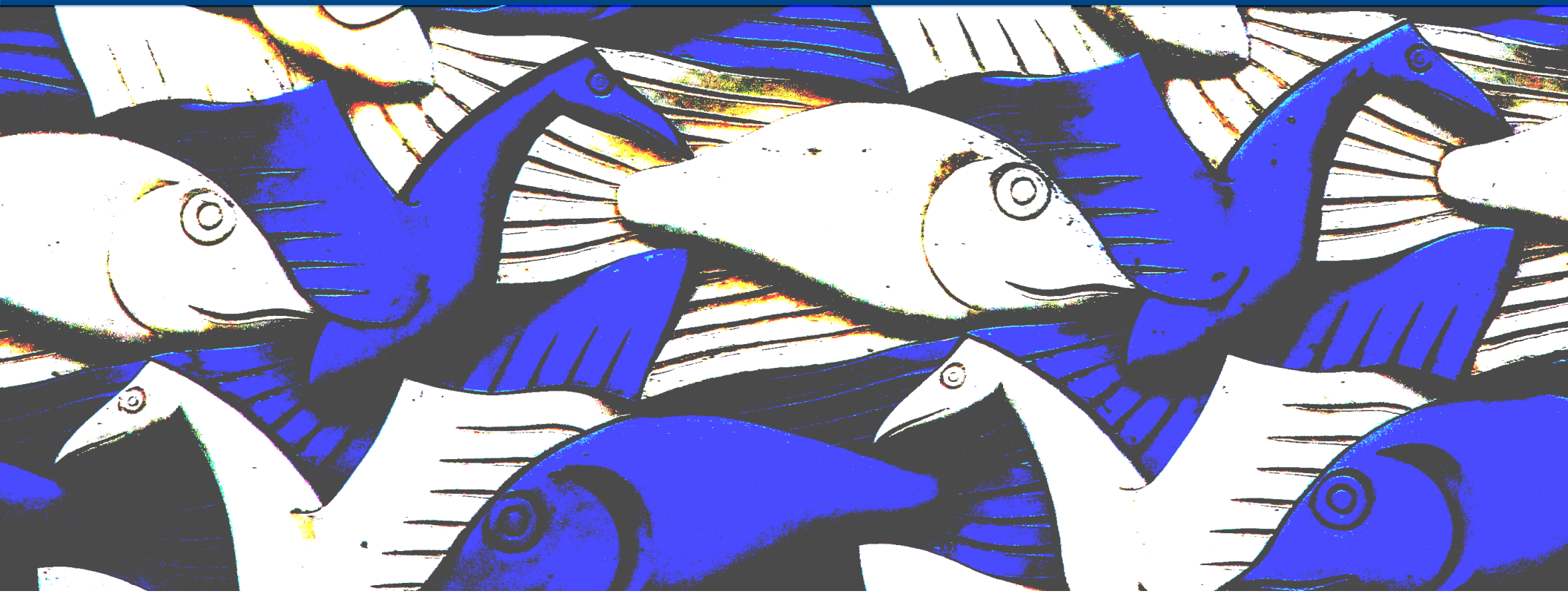
Recherche WEB



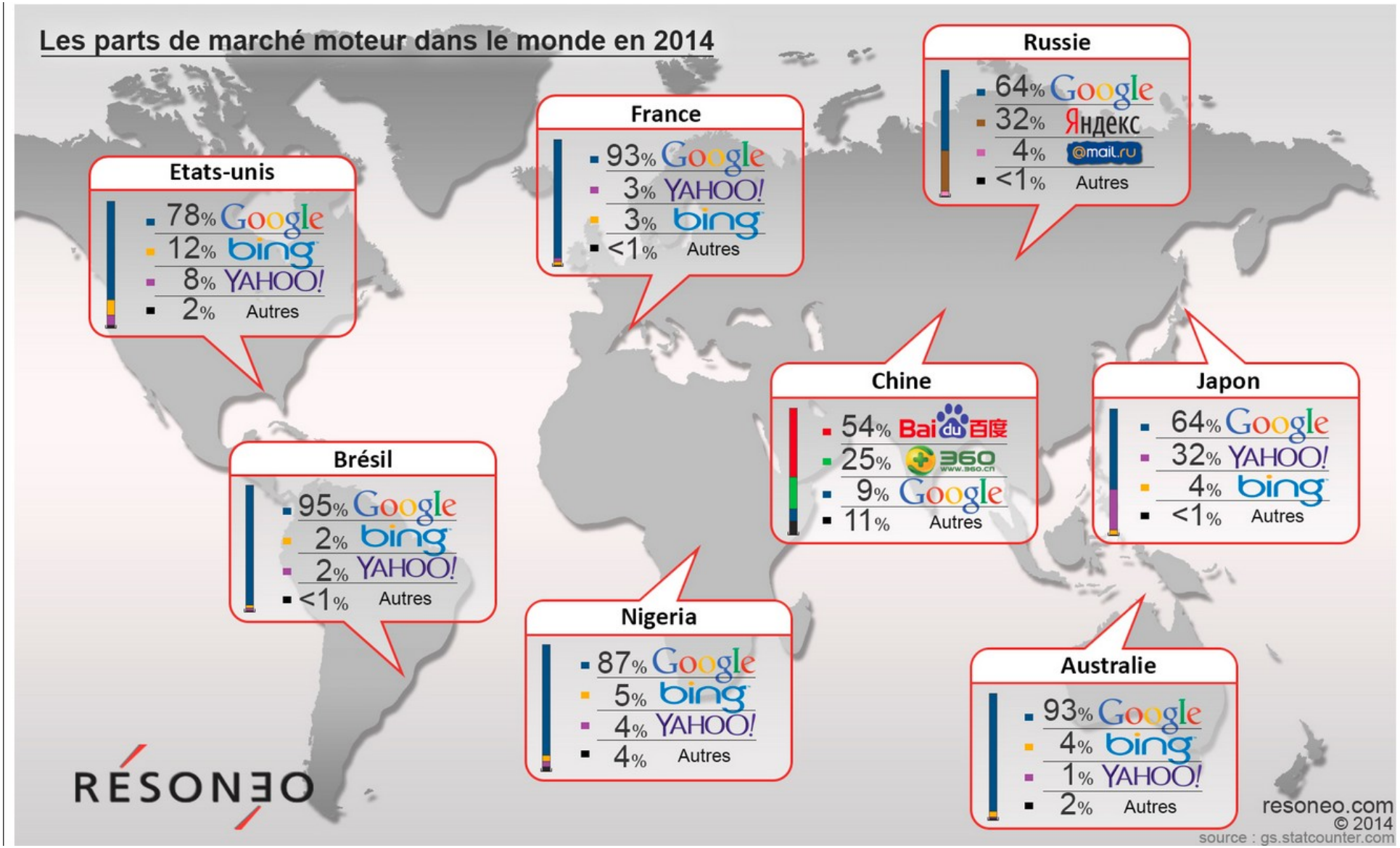
Raisonnement et science de la décision

Master M1 MIAGE – Ingénierie Métier

Université Toulouse 1 Capitole

Umberto Grandi

Les moteurs de recherche dans le monde



Pourquoi chercher est-il difficile ?

Jusqu'aux années 80 les algorithmes de recherche (search, information retrieval...) était fait pour des archives composées d'articles de journaux, des brevets, d'articles scientifiques...

Le problème principal était donc la
pénurie des sources d'information
(d'ici le mot « retrieval », récuperation)

Pourquoi chercher est-il difficile ?

Le principaux instruments de recherche étaient les **mots clés**

Deux les problèmes principaux :

1)La **synonymie** : multiple façons de dire la même chose. Par exemple si je cherche des « patates » je veux trouver des « pommes de terre »

2)La **polysémie** : multiple significations de la même expression. Par exemple si je cherche « jaguar » suis-je intéressé par l'animal ou le constructeur de voitures ?

Difficultés de chercher sur le WEB

Problèmes spécifiques
à la recherche WEB :

- Information dynamique
- Surabondance des documents
- Fiabilité des sources
- ...

Exemple : Le 11 Septembre 2011 beaucoup d'utilisateurs cherchaient «world trade center » sur Google. Mais ce qu'ils trouvaient était des description du bâtiment, car l'indexation des sites n'était faite que périodiquement…

La recherche d'information doit donc prendre en compte cette nouvelle **dynamique** !

Difficultés de chercher sur le WEB

Chaque utilisateur d'Internet est aussi producteur d'information (Facebook, blogs…) : on est passé de chercher une aiguille dans une botte de foin, à un problème de **sélection des sources** !

Dans ce cours on verra comment utiliser la structure du WEB pour obtenir un classement des pages selon leurs fiabilité ou autorité

Contenu du cours

Le concept clé pour les algorithmes de recherche WEB qu'on verra est de ne regarder que la **structure des liens** qui relient les pages, **pas leur contenu**

On étudiera deux algorithmes fondés sur l'analyse des liens :

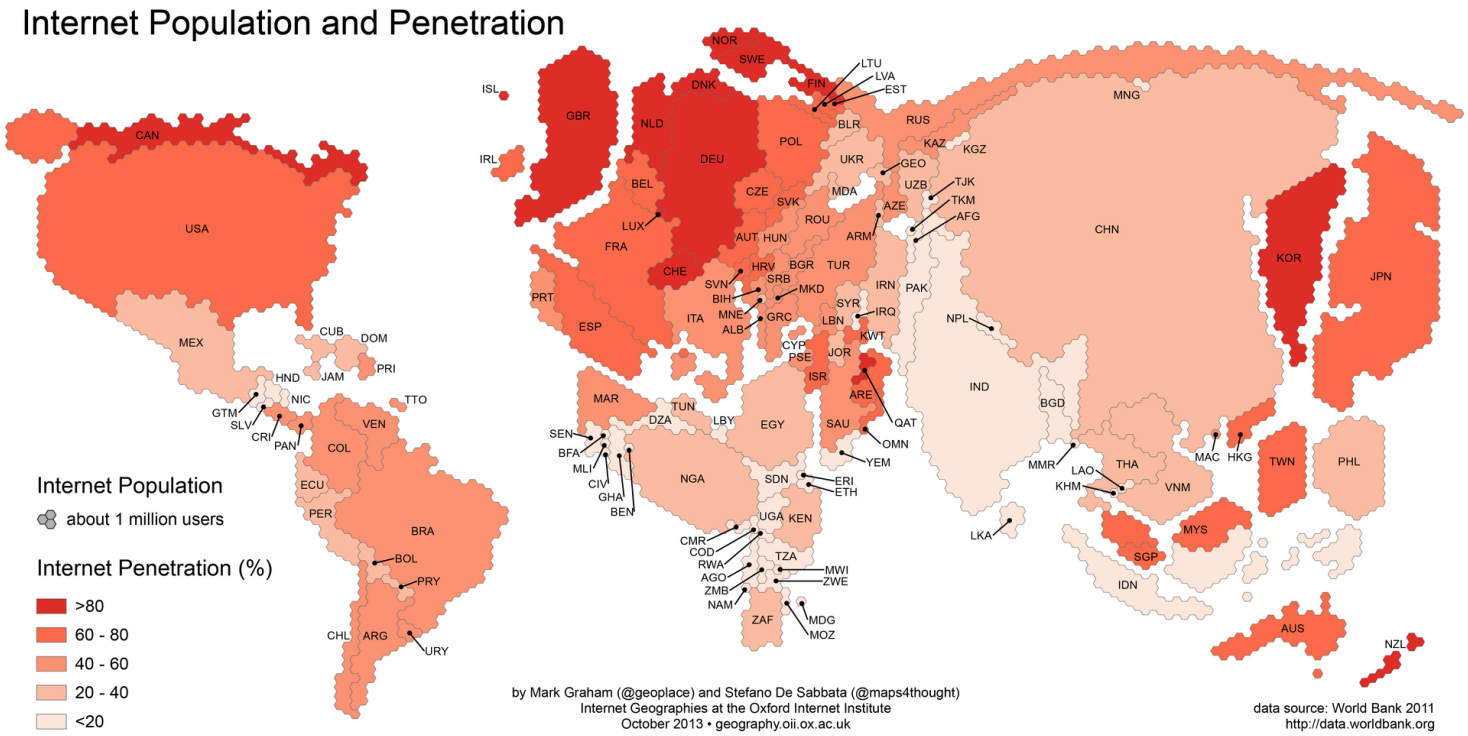
1) Hubs and authorities

2) Page Rank

Internet et le World Wide Web

Internet est le **réseau informatique mondiale**, un réseau de réseaux, composé des cables, transmetteurs d'ondes, serveurs...et des protocoles pour la transmission d'information (IP, TCP, HTTP, FTP...)

Evolution d'ARPANET, conçue dans les années 60.



Internet et le World Wide Web

Le World Wide Web (« la toile », le WEB) est une des applications d'Internet :

- Emails
- Peer-to-peer
- Chats
- ...
- **Hypertexte et pages Internet**

Inventé par Tim Berners-Lee et autres chercheurs du CERN au début des années 90.

Exercice : écrire sur papier un fragment de la toile en partent de page d'accueil de l'IRIT : www.irit.fr

Hubs and authorities 1.0

Un des plus simples algorithmes de recherche WEB est le suivant :

Algorithme de vote par liens entrants :

1)Faire une **recherche par mots clés** et obtenir toutes les pages qui parlent d'un sujet (exemple: tous pages avec le mot clé « Capitole »)

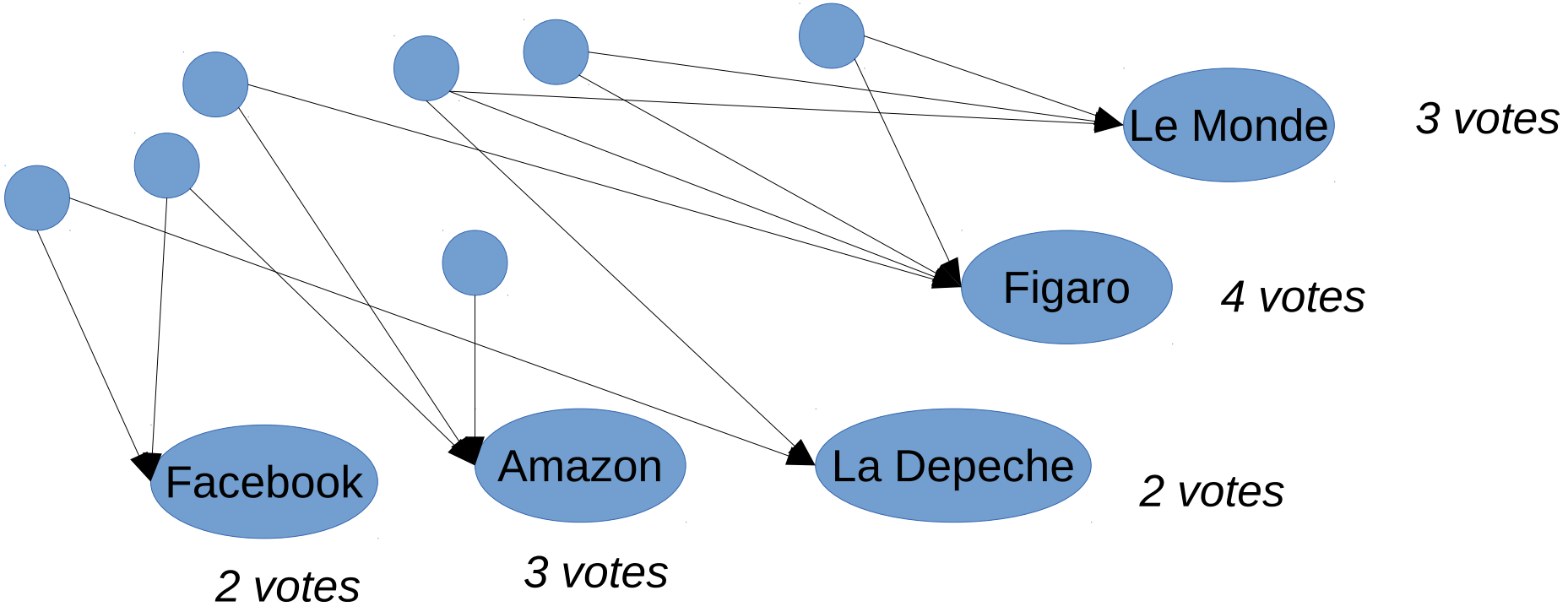
2)Extraire la **fraction du WEB** (un réseau) qui connecte ces pages

3)Classer ces pages par nombre de liens entrants : **plus** une page est reliée par d'autres pages, **plus** cette page est importante

Le vote par liens entrants est un algorithme qui marche bien quand on recherche une **seule page**. Quoi faire si on cherche une **liste**, comme « universités Toulouse » ?

Listes de sites

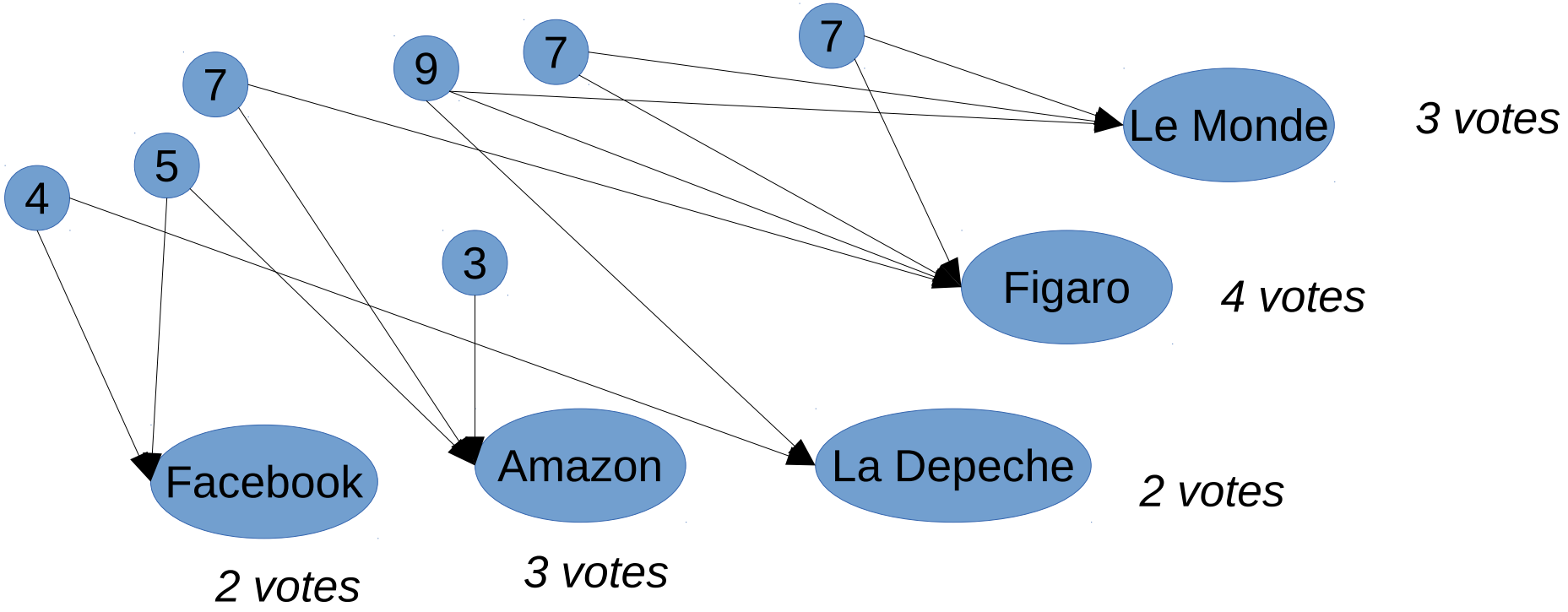
Une recherche classique donne comme résultats des pages comme Facebook ou Amazon, qui reçoivent des liens entrants qui ne sont pas pertinents pour notre requête. Par exemple , si on cherche « quotidiens » :



On voudrait plutôt identifier **des pages qui contiennent une liste des liens** vers les objets de notre requête - par exemple une liste de quotidiens. Mais cela peut être lu directement à partir de la structure du WEB :
P*ouvez-vous identifier ces pages dans la figure ci-dessus ?*

Le principe d'amélioration répété

Les sources qui recommandent les pages avec le plus de votes sont les plus fiables



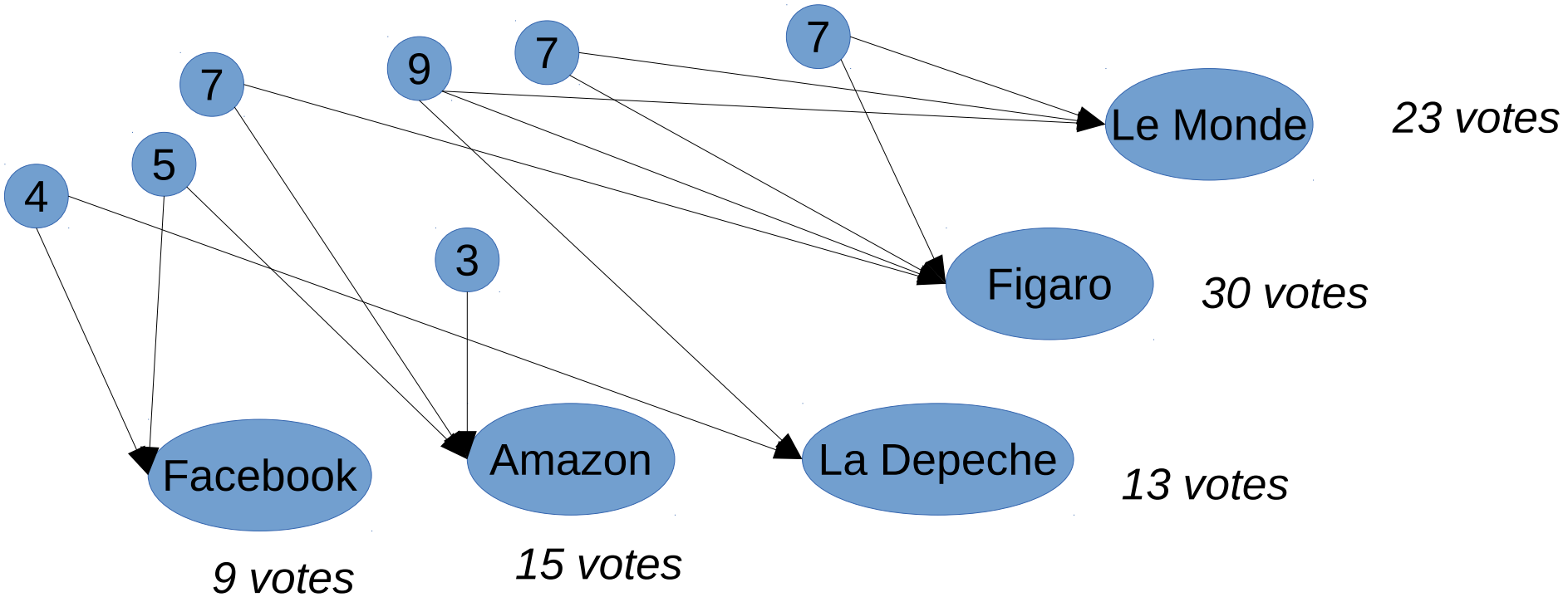
On peut associer à chaque page source **la somme des points** des pages objets

Le principe de l'amélioration répété

Pourquoi s'arrêter là ?

Les pages recommandées par des pages à haute fiabilité sont les plus importantes

On peut donc recalculer l'importance des pages objets (les quotidiens) à partir de la fiabilité des sources qu'on vient de calculer :



Hubs and authorities 2.0

Algorithme de « hubs and authorities » (pour k pas)

Objectif : Calculer pour chaque page *p* (obtenues par exemple avec une première recherche par mots clés) un score de hub, qu'on notera *hub(p),* ainsi que une score de authority *auth(p)*

1)Initialiser *hub(p)=auth(p)=1* pour toutes pages *p*

2)Choisir un nombre des pas *k*

3)Répéter *k* fois les deux règles de mise à jour suivantes :

a)Authority update : Pour chaque page *p*, mettre à jour *auth(p)* à la somme des scores *hub* des pages qui pointent vers *p.*

b)Hub update : Pour chaque page *p*, mettre à jour *hub(p)* à la somme des scores *auth* des pages vers qui *p* pointe.

Normaliser les deux scores obtenus. *(Exercice : qu'est que ça veut dire?)*

4)Choisir la page avec le score de *auth* le plus élevé.

La transitivité de l’approbation

On peut observer que dans plusieurs domaines de recherche, l’approbation d’une page vers une autre est transitive :

Si une page *p* est nommée par une page réputée, la page *p* sera aussi réputée

Cette observation, combinée avec les votes sur les liens entrants, a donné naissance au célèbre algorithme de **PageRank**

PageRank 1.0

On peut imaginer le score de PageRank d'une page comme un liquide qui circule dans le WEB et se concentre aux nœuds les plus importants :

Algorithme PageRank 1.0 :

1)Dans un réseau de *n* pages WEB, initialiser le PageRank de chaque page à *1/n*

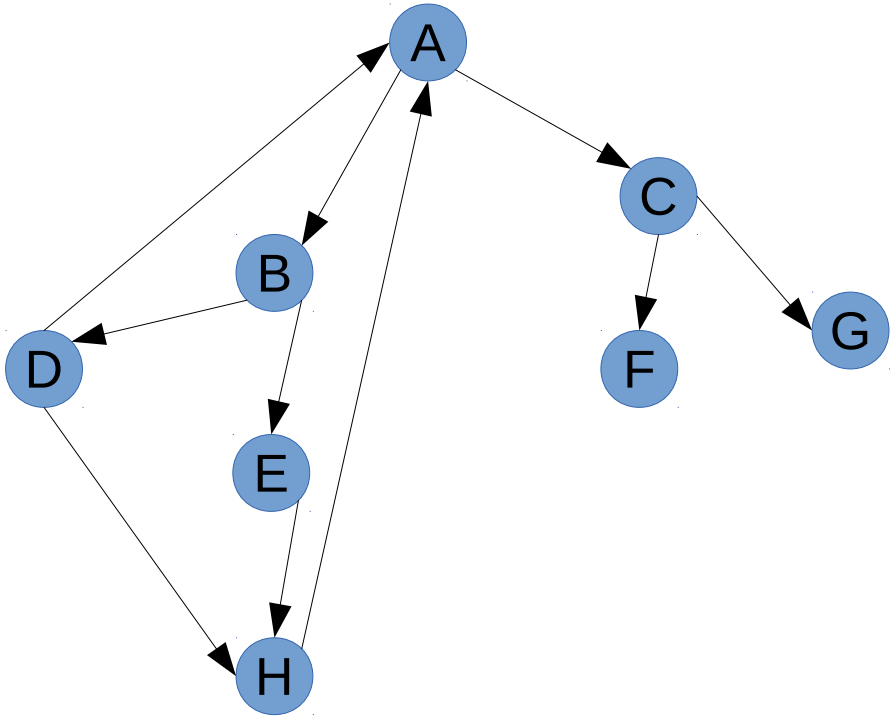
2)Choisir un nombre des pas *k*

3)Répéter k fois la règle de mis à jour de PageRank :

PageRank update 1.0:

- Chaque page divise son PageRank par le nombre de liens sortant
- Envoie un « paquet » de PageRank de cette valeur aux pages vers lesquelles elle pointe (si une page n'a pas de liens sortants elle envoie tout son PageRank à elle même)
- Chaque page met à jour son PageRank à la somme des « paquets » de PageRank reçus par les autres pages

Exemple : PageRank 1.0



Toutes pages commencent avec 1/8

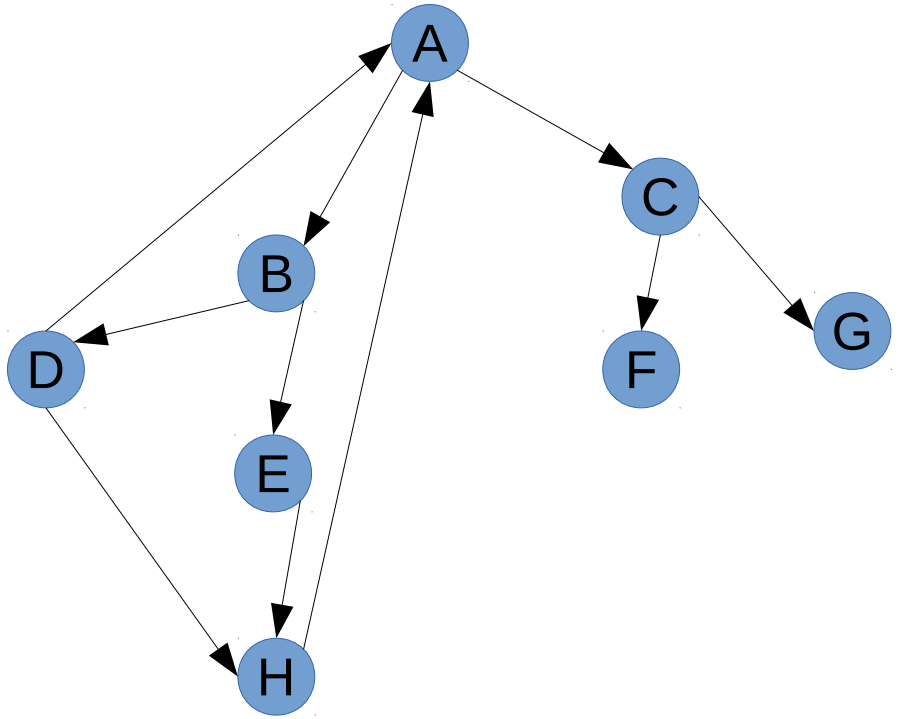
	Pas 0	Pas 1	Pas 2
A	1/8	3/16	7/32
B	1/8	1/16	3/32
C	1/8	1/16	3/32
D	1/8	1/16	1/32
E	1/8	1/16	1/32
F	1/8	3/16	7/32
G	1/8	3/16	7/32
H	1/8	3/16	3/32

Observation : la somme des PageRank est toujours égale à 1

PageRank 1.0

L’algorithme de PageRank 1.0 souffre d’un problème grave : le score de PageRank **se concentre** chaque fois qu’il y a une **impasse**, car ce score ne peut pas etre transféré ailleurs...

Exemple : les nœuds F et G dans le réseau ci-dessous



PageRank 2.0

La solution est d'ajouter à la règle de mise à jour un moment de redistribution du score Page Rank :

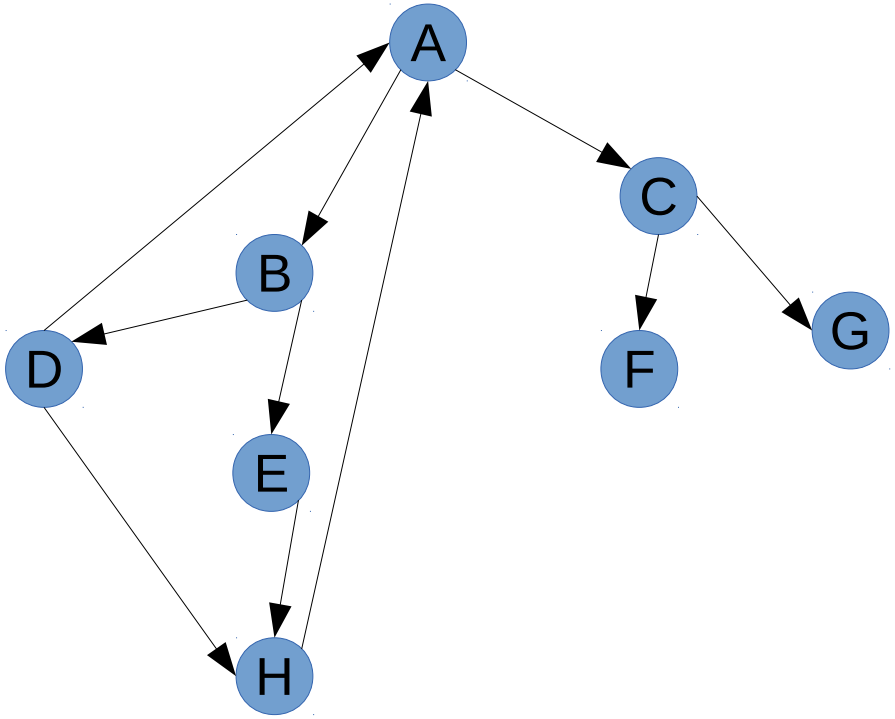
PageRank update 2.0

- Appliquer la *PageRank update 1.0*
- Multiplier toutes scores PageRank par un facteur d'actualisation $s<1$. Cela veut dire que le PageRank totale passe de 1 à s.
- Diviser également entre tous page le score de $1-s$ qui reste, en ajoutant $(1-s)/n$ au score PageRank de chaque page.

À savoir que :

- Une fois fixé le réseau, si l'on répète à l'infini cette mise à jour, on convergera vers un ensemble de **scores PageRank unique**
- Les valeurs du facteur d'actualisation s sont entre 0,8 et 0,9

Exemple : Page Rank 2.0



Facteur d'actualisation de 0,9
Redistribution de 0,0125 à chaque pas

	Pas 0	Pas 0.1	Pas 1
A	1/8	0,169	0,181
B	1/8	0,056	0,069
C	1/8	0,056	0,069
D	1/8	0,056	0,069
E	1/8	0,056	0,069
F	1/8	0,169	0,181
G	1/8	0,169	0,181
H	1/8	0,169	0,181

Observation : l'effet de la redistribution est apparent après plusieurs pas

La pratique

Les algorithmes de recherche utilisés par la plupart de moteurs de recherche sont très très **secrets**. La plupart n'utilise *PageRank* qu'en partie, à cause des avancement des logiciels, notamment en traitement automatique du langage et en apprentissage automatique. L'algorithme de *hubs and authoritites* est lui aussi très utilisé.

La recherche WEB de nos jours

1) Techniques de recherche plus sophistiquées : par exemple développer des logiciels pour lire et comprendre le **texte ancre** dans un lien.

   Bon exemple : cliquer ici pour le site de l'Université Toulouse 1 Capitole
   Mauvais exemple : cliquer ici pour le site de l'Université Toulouse 1 Capitole

2) Comprendre les **stratégies** des pages WEB :

   a) Le **référencement** est trop important pour les créateurs de pages WEB. Ils tentent toutes stratégies pour être premier dans les résultats de recherche. « *La recherche WEB est une application d'information retrieval aux documents qui se comportent mal* » (Cliff Lynch)

   b) Garder secret l'algorithme de recherche

3) Transformer la recherche WEB dans un modèle de business fondé sur la **publicité**. Les premiers résultats d'une recherche sont alloués et vendus avec des mécanismes d'**enchères** très sophistiqués

Liens avec d’autres cours du Master IM

- *Conception de sites Web* : le référencement d’une page est crucial, et connaître les algorithmes de recherche est un avantage
- *Programmation structurée* : nous avons introduit plusieurs algorithmes. Pouvez-vous les écrire dans un langage de programmation ? Reconnaissez-vous la « forme » de ces algorithmes ?
- *Bases de données* : Quelles sont les différences et les parallèles entre les algorithmes de recherche WEB et les requêtes SQL ?
- Dans la dernière partie de ce cours on étudiera les mécanismes d’enchère utilisés par les moteurs de recherche pour allouer les espaces publicitaires.

Références

Ce cours est basé sur le matériel suivant :

- *D. Easley and J. Kleinberg. Network, Crowds and Markets. Cambridge University Press, 2010. Chapter 14*

Le célèbre algorithme de PageRank était inventé par Larry Page et Sergey Brin, co-fondateurs de Google, dans l'article suivant :

- *Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In Proceedings of the 7th Internation World Wide Web Conference, 1998.*