Le fonctionnement de Google™

Etude de la structure du World Wide Web

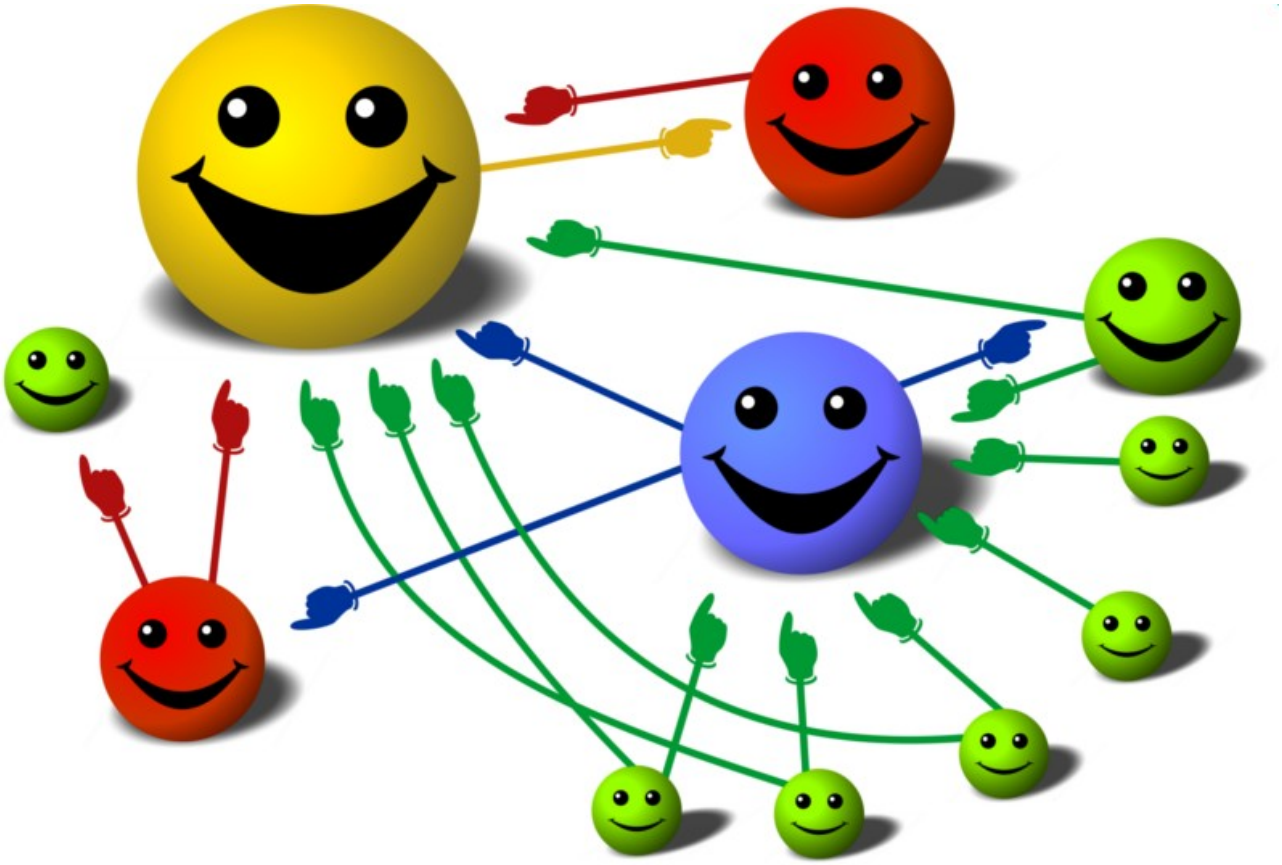


Image : Felipe Micaroni Lalli pour Wikipedia

Bourse Yoann - Lemaire Alan

Plan du dossier

Consultez les annexes

L'expansion d'internet a été une véritable révolution dans le monde des communications, et sans doute l'évènement le plus marquant de ce début de millénaire. Un nombre exponentiellement croissant d'individus a pu être connecté à un **réseau mondial d**'envergure inimaginable. Dès lors, des serveurs se sont formés pour proposer du contenu. Des milliards de documents ont bénéficié de ce mode d'échange d'informations révolutionnaire : les fameuses pages web. Chacun a bientôt pu apporter sa propre pierre à ce qui semble s'imposer comme **la plus grande base de données de tous les temps**. En quelques années, la toile a pris des proportions astronomiques. Mais ce formidable outil serait resté bien vain si rien n'avait été fait pour y voir clair dans cet enchevêtrement démesuré. Heureusement, des **moteurs de recherche** se sont penchés sur cette question cruciale.

Les estimations suggèrent que la toile compte actuellement plus de mille milliards de pages web (*Jesse Alpert & Nissan Hajaj, official Google Blog, 25/7/08*). A titre de comparaison, ceci représente à peu près le nombre de brins d'herbe dans 10 kilomètres carrés de prairies, ou le nombre de grains de sable dans un cube de 10 mètres d'arête. Il semble alors logique que, lorsqu'un utilisateur recherche des informations sur les voitures par exemple, le nombre de pages faisant apparaitre le mot “voiture” soit tout simplement énorme. L'enjeu de la fouille dans cette base de donnée si particulière par ses proportions titanesques est bien sur d'**ordonner cette liste de résultats pour permettre à l'utilisateur du moteur de recherche d'accéder le plus facilement et le plus rapidement possible à l'information requise**. C'est cette capacité à répondre au mieux à la demande qui fait la force d'un moteur de recherche.

Pour optimiser ces résultats, il est donc nécessaire de bien connaitre la toile et sa structure. A la fin des années 1990, une petite firme de Mountain View a bien compris cet enjeu. Son produit, le célebrissime moteur de recherche **Google**, a connu un succès tout simplement fulgurant, s'imposant très rapidement comme le meilleur de sa catégorie. Son secret ? **Un algorithme révolutionnaire connu sous le nom de Pagerank, qui permet une hiérarchisation des résultats de recherche basée sur la structure très particulière du web.**

En effet, la toile du web n'est pas un réseau aléatoire. Les pages web, ces documents dont nous avons déjà parlé, sont **connectées** entre elles par d'innombrables **liens hypertextes**, des mots cliquables qui dirigent l'internaute vers d'autres sites. Mais on remarque rapidement que ces liens ne sont pas répartis au hasard. En effet, il existe des **pages beaucoup plus populaires** que d'autres auxquelles mènent beaucoup plus de liens. Ce sont ces sites web qui sont véritablement le coeur d'internet.

C'est cette structure que Google a su mettre a profit. Son algorithme dote chaque page d'une valeur de popularité, désignée par le nom Pagerank, les plus populaires étant bien entendu les pages vers lesquelles mènent le plus de liens. Face à sa réussite exceptionelle, la firme de Mountain View **garde jalousement les secrets** de son moteur de recherche. Nous avons essayé de les approcher.

Pour une bonne compréhension de cette présentation, quelques définitions doivent être précisées.

Internet

C'est le réseau informatique mondial qui permet l'utilisation du courier électronique ou du World Wide Web. C'est en quelques sortes une immense plateforme à laquelle de nombreux ordinateurs dans le monde entier sont connectés.

World Wide Web (ou Web)

Le World Wide Web (*"la toile mondiale"*) est le service le plus célèbre permis grâce à internet, avec qui il est souvent confondu. Il s'agit d'un ensemble de documents consultables via internet, reliés entre eux par des liens appelés liens hypertextes.

Page Internet / Page Web

C'est le nom couramment utilisé pour désigner les documents présents sur le World Wide Web. Chaque page est repérée par une adresse appelée URL (*Uniform Resource Locator*).

Site Internet / Site Web

Un site web est un ensemble de pages web, souvent du même auteur, particulièrement liées entre elles. C'est en quelques sortes l'équivalent d'un dossier.

Lien hypertexte

Les pages internet sont inter-connectées par des liens hypertextes : ce sont des portions de texte ou des images sur lesquelles l'internaute peut cliquer pour être dirigé vers une autre page. Ce sont les connexions qui relient tout le réseau du World Wide Web.

Moteur de recherche

Il s'agit d'un site web particulier qui propose à l'internaute de l'aider à débusquer des documents dans l'immense World Wide Web. Il fouille dans cette base de donnée pour fournir à l'utilisateur une liste de pages web sur lesquelles il pourrait trouver les informations qui l'intéressent.

Avant-Propos

Tri d'une base de données

La toile mondiale du web contient à l'heure actuelle plus de **mille milliards de pages web**, et ce nombre continue d'augmenter en permanence. C'est sans contexte la plus grande base de données jamais connue, et probablement la plus utilisée.

Les seuls moyens de se repérer dans cet immense réseau sont les **moteurs de recherche**, des sites web qui fouillent la toile entière pour aider les internautes à trouver les informations désirées.

Le fonctionnement de ces petits robots est simple : **l'usager n'a qu'à entrer un ou quelques mots clefs, et le programme se charge de lui retourner la liste des sites web contenant sa requête.**

En théorie, l'internaute devrait ainsi arriver directement sur les sites qui l'intéressent. Dans la réalité, tout n'est pas aussi simple. Une recherche, aussi pointue soit-elle, aboutit à une **quantité incroyable de résultats**, dans une base aussi grande. Pour donner quelques exemples chiffrées, si 16 millions de pages comportent le mot “carré”, près de 313 millions contiennent “square”, sa traduction anglaise. L'expression “endomorphisme”, beaucoup moins courante, est présente sur près de 30 mille sites (*Chiffres Google, septembre 2008*). Le nombre de résultats est en tout cas impressionnant, et ils ne peuvent donc pas être présentés tous simultanément à l'utilisateur. De plus, celui-ci perdrait un temps considérable à parcourir toutes les pages. Il est donc important de lui présenter les résultats qui conviennent le mieux à sa demande.

Se pose alors la question d'**organiser les résultats**. Aucun robot n'est assez performant pour égaler la capacité de compréhension de l'esprit humain, et personne ne peut répondre assez rapidement aux requêtes et trier un si grand nombre de documents. Il faut donc paramétrer les robots assez efficacement pour proposer un programme utile.

Pour mieux comprendre les enjeux de ce problème, observons les résultats d'une requête sans aucun tri. En recherchant “Théorème de Heine”, on peut bien entendu trouver l'énoncé de ce théorème de mathématiques. Mais tout le monde n'aura pas cette chance : entre les nombreuses fausses pages web qui présentent des listes de mots clés pour attirer les visiteurs malencontreux sur leurs publicités et les requêtes qui ne conviennent pas, l'utilisateur prendra peut-être un long moment avant de trouver ce qu'il cherche. Comme il serait lourd de vous présenter tous les résultats erronés ici, nous nous contenterons d'un court échantillon.

Forum d'analyse - Les Mathématiques.net
On répète de nombreuses fois l'expression “Théorème de Heine” simplement pour expliquer que ce n'est pas la méthode à utiliser dans l'exercice dont il est question.
http://les-mathematiques.u-strasbg.fr/phorum5/read.php?4,455414,455680

Forum de discussion - Les Mathématiques.net
Encore une fois, le terme “Théorème de Heine” est évoqué au cours d'une discussion entre ingénieurs évoquant leurs parcours professionnel.
http://les-mathematiques.u-strasbg.fr/phorum5/read.php?6,394006,page=1

Descriptif de H-Prépa Maths. Analyse 1
Le Théorème de Heine fait partie de la liste des notions que le livre aborde et qui sont proposées dans cette publicité.
http://publimath.irem.univ-mrs.fr/biblio/M1U99026.htm

Wikipedia, Théorème d'Ascoli
La mention du Théorème de Heine figure dans la démonstration de ce théorème.
http://fr.wikipedia.org/wiki/Th%C3%A9or%C3%A8me_d%27Ascoli

La grande majorité des résultats ne donne ainsi aucune indication quant au contenu du théorème et se révèle donc tout simplement inutilisable. Cependant, l'expression étant peu commune, les résultats restent en grande partie limités au domaine des mathématiques. Ce ne serait pas le cas avec des mots plus courants. Imaginons qu'un internaute souhaite se procurer un oreiller. Il pourrait tomber sur des résultats aussi variés que :

Gâteau “oreiller”

La recette d'un gâteau portant le nom de l'objet.

http://zapbook.canalblog.com/archives/2008/09/28/10751834.html

Citations oreiller

Un recueil de citations d'œuvres littéraires comportant le mot oreiller.

http://www.dicocitations.com/citation.php?mot=oreiller

Confidences sur l'oreiller

Un site internet traitant de la télévision réalité.

http://secret-story.tf1.fr/videos-photos/videos-jour/confidences-sur-oreiller-3908854.html

Pour un allaitement réussi

Ce livre virtuel traite de l'accouchement, et il y a mention d'un oreiller.

http://books.google.fr/books?id=Lf-ATn1B8cgC&pg=PA17&lpg=PA17&dq=oreiller&source=web&ots=5ZyutDjkc0&sig=PfqiJWFeOkYB724jQGDk_s_VBYk&hl=fr&sa=X&oi=book_result&resnum=3&ct=result

Dictionnaire de l'Académie Françoise

Ce dictionnaire de 1765, numérisé par la firme Google, offre une définition du terme.

http://books.google.fr/books?id=Osjzy4S5JCQC&printsec=titlepage&dq=oreiller&source=gbs_summary_s&cad=0

Tout se règle sur l'oreiller

Cette page personnelle traite de disputes conjugales et d'éducation sexuelle.

http://lesimpatiencesamoureuses.over-blog.com/categorie-1067171.html

Un autre exemple tout aussi frappant avec la recherche du terme « monôme », pourtant assez peu polysémique.

Monôme
Une firme anglophone commercialisant de l'électronique
http://www.monome.org
Une manifestation étudiante de la fin du XIXe siècle
http://fr.wikipedia.org/wiki/Monôme
Un article de presse intitulé "le monôme des socialistes"
www.mediapart.fr/club/blog/jean-michel-helvig/280708/le-monome-des-socialistes
Un article traitant de cryptographie
www.apprendre-en-ligne.net/crypto/subst/monobi.html
Un blog personnel
http://monome-fourcade-2008.skyrock.com/
Un livre "Le Grand Monome" de Yves Gibeau en vente aux enchères
www.priceminister.com/offer/buy/3109059/Gibeau-Yves-Le-Grand-Monome-Livre-ancien.html

Ainsi, l'utilisateur peut perdre un **temps considérable** à parcourir l'immense liste des résultats en quête de ce qu'il cherche. Le moteur de recherche ne peut pas lui présenter en même temps les nombreuses pages qui contiennent le mot recherché. De plus, s'il n'y a aucune sélection, les sites retournés pourraient s'avérer être de **mauvaise qualité, incomplets ou erronés, voire pire : frauduleux ou illégaux**. L'utilisateur pourrait rencontrer des marchands peu scrupuleux qui pourraient abuser de sa confiance par divers biais et tromperies. Un filtrage effectué par des humains serait draconien, en raison de la multitude de documents à trier. De plus, cela ne garantirait pas forcément la **cohérence ou la pertinence des résultats**, comme nous l'avons vu ci-dessus.

D'où **la nécessité d'un classement efficace, privilégiant les sites de confiance, qui proposent les informations les plus claires et les plus directes**. C'est ce critère qui a rendu populaire le moteur de recherche le plus utilisé au monde, Google. Quand ses concurrents se contentaient de proposer des listes de résultats non triés, les ingénieurs de la firme de Mountain View ont mis au point un algorithme qui allait changer à tout jamais la face du monde virtuel. Leurs recherches sur la structure d'internet les ont poussés à exploiter au mieux ce réseau si particulier. Le succès fulgurant du programme en résultant confirma la véracité de leurs théories.

Partie I

Le système « Pagerank »

Sergei (Mikhaïlovitch) Brin 1973-
Larry (Lawrence E.) Page 1973-
Cofondateurs et codirecteurs actuels de Google

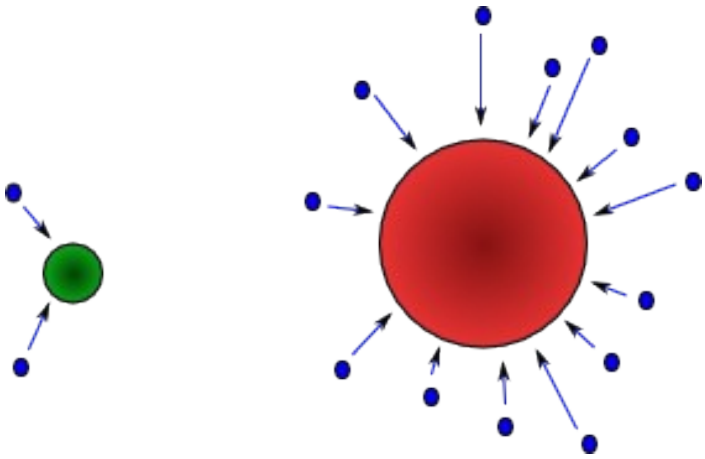
Pour trier les listes de résultats, l'idée révolutionnaire de Google fut d'attribuer à chaque site **une note, qu'on appelle "Pagerank"**. C'est en quelques sortes **un indicateur de confiance**. Il classe donc naturellement ses résultats par pagerank décroissant.

Le nom "Pagerank" a un double sens. Littéralement "Classement de la page", c'est également un hommage à Larry Page, cofondateur de Google et inventeur de ce principe.

Mais un problème se soulevait alors : comment attribuer un pagerank à chaque page ? Fallait-il déployer une quantité colossale de salariés qui parcourrait la toile sans relâche pour noter chaque page ? Ce projet semblait peu réaliste. Il fallait un algorithme simple et autonome, qui puisse être utilisable par des robots et réactualisé en permanence.

La solution vint de leurs études poussées de la structure du web. Leur idée était très simple :

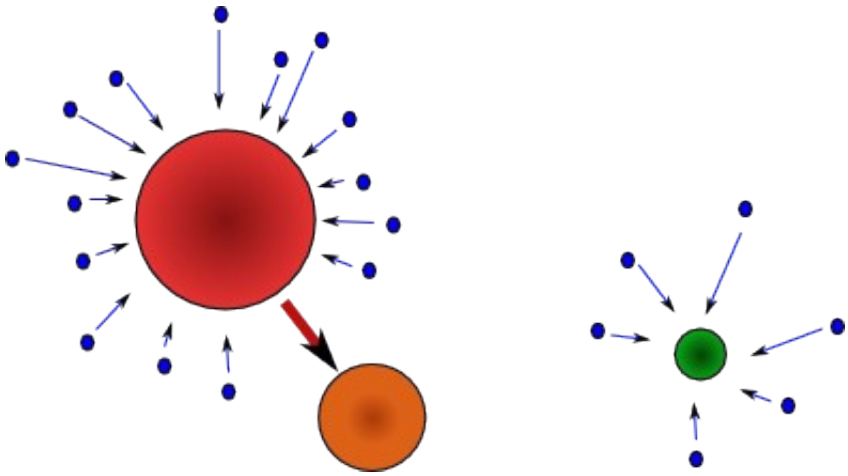
Plus un site est fiable et complet, plus il est populaire, et plus il est donc cité sur d'autres sites ou forums de discussion. Ainsi, plus le nombre de liens pointants vers le site est important, plus celui-ci doit être fiable.



Ce principe produisit des résultats étonnamment adéquats. Il était cependant perfectible.

Chaque lien vers un site est considéré comme un plébiscite en faveur de ce dernier. Mais compte tenu de la structure du world wide web, et de la hiérarchisation qu'on veut lui appliquer, il semble alors logique d'ajouter une seconde règle.

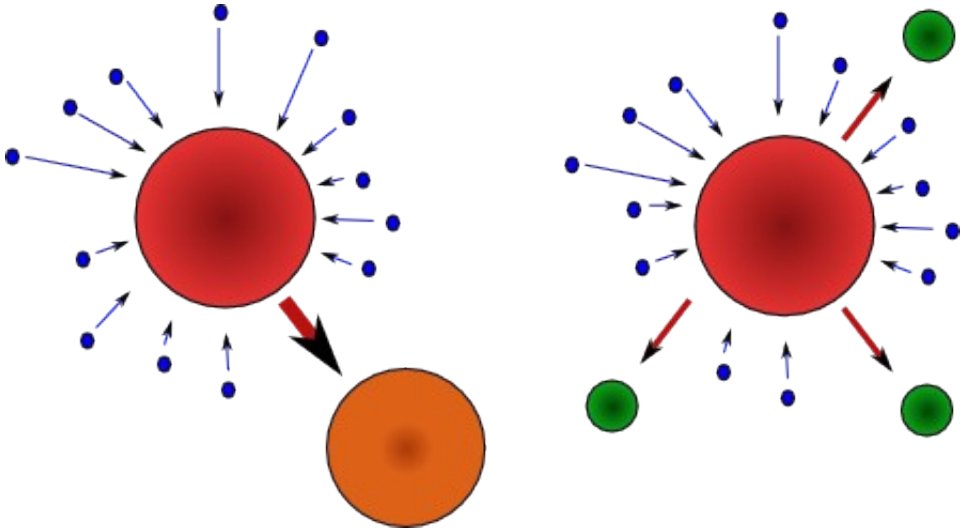
La contribution au pagerank d'un lien pointant vers un site est d'autant plus grande que le pagerank du site d'où vient le lien est important.



Ainsi, chaque lien à en quelques sortes une valeur, le pagerank du site d'où il provient. Un lien provenant d'un site à fort pagerank, un site de confiance, sera considéré comme fiable et augmentera de beaucoup le pagerank de sa cible, parfois plus qu'une multitude de liens provenant de sites à faibles pageranks, donc moins fiables.

Pour finir, par analogie avec un plébiscite classique,

La contribution au pagerank d'un lien pointant vers un site est d'autant plus grande que le nombre de liens provenant de la même source est faible.



Ce qui permet d'éviter les abus, et de tenir compte du fait qu'un vote est plus précieux s'il est unique. Il existe en effet divers annuaires de sites, qui présentent de véritables listes de sites, et dont il semble normal que la contribution au pagerank soit réduite.

Ces deux principes sont fondateurs de l'algorithme à la source de la renommée de Google, celui du calcul du pagerank. Il se décrit mathématiquement de la façon suivante.

Considérons une page A auquel mènent q liens de q pages différents. On notera P_i le pagerank de la page i, et S_i le nombre de liens sortants de la page i. La formule donnant le pagerank du site A sera :

$$P_a = [1 - c] + c\left(\frac{P_1}{\text{degré}(1)} + \frac{P_2}{\text{degré}(2)} + \ \ldots\ + \frac{P_q}{\text{degré}(q)}\right)$$

Où c est un coefficient estimé aux alentours de 0,25.

L'utilisation du coefficient c se justifie par un modèle probabiliste. En effet, on peut imaginer un internaute fictif qui se déplacerait sur internet. A chaque itération du calcul, on considère qu'il se déplace de site en site. c correspond à la probabilité qu'il change de page, et donc $[1 - c]$ à la probabilité qu'il reste sur la même page. Il choisit entre tous les liens qui lui sont offerts selon une probabilité proportionnelle au Pagerank du site de destination, ce qui lui donne autant de possibilités que le nombre de liens sur la page source, soit son degré. En sommant les quotients, on obtient ainsi toutes les possibilités pour l'internaute d'atteindre la page A . Le Pagerank de la page A est alors augmenté par un déplacement vers elle du rapport du Pagerank de la page d'origine sur le degré sortant de cette dernière.

D'autre part, le coefficient c est obligatoire pour assurer la convergence de la méthode itérative. En effet, nous pouvons traduire cette égalité sous la forme d'un calcul matriciel dont nous montrons en page suivante qu'il converge. Il s'agit de traduire le processus itératif par l'application répétée d'une matrice à un vecteur dont chaque composante comporte le Pagerank temporaire d'un site. La formule du Pagerank devient alors, sous forme matricielle :

$$T(x) = c\,\varepsilon + (1 - c)\,A\,x$$

avec :

- ***x*** : vecteur dont chaque coordonnée représente la probabilité de présence sur une page (*son pagerank*).

- ***c*** : probabilité de changer de page.

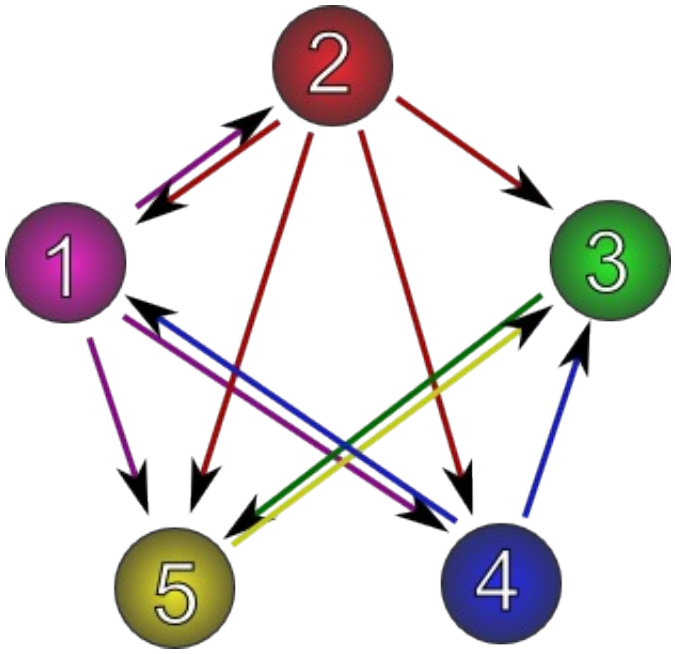
- ***A*** : matrice contenant les liens entre les sites web

si le site *j* présente un lien vers le site *i* : $A_{i,j} = \dfrac{1}{\text{degré}(i)}$

sinon : $A_{i,j} = 0$

Exemple :

	de 1	2	3	4	5
vers 1	0	$\frac{1}{4}$	0	$\frac{1}{2}$	0
vers 2	$\frac{1}{3}$	0	0	0	0
vers 3	0	$\frac{1}{4}$	0	$\frac{1}{2}$	1
vers 4	$\frac{1}{3}$	$\frac{1}{4}$	0	0	0
vers 5	$\frac{1}{3}$	$\frac{1}{4}$	1	0	0



Il s'agit de montrer que l'application $T(x) = c\,\varepsilon + (1 - c)\,A\,x$ *de* R^n *dans* R^n *est une application contractante, avec* **c ≈ 0,25**

$$z = T(x) - T(y) = (1 - c)\,A\,(x - y)$$

Il aura pour coordonnées :

$$z_i = \sum_{k=0}^{n} (1 - c)\,A_{i,k}\,(x_k - y_k)$$

D'où les inégalités :

$$\|z\|_1 = \sum_{i=0}^{n} |z_i| \le \sum_{i=0}^{n} \left(\sum_{k=0}^{n} (1-c)\,|A_{i,k}|\,|x_k - y_k| \right)$$

Or $\sum_{i=0}^{n} |A_{i,k}| = 1$ d'où $\|z\|_1 \le (1-c) \sum_{k=0}^{n} |x_k - y_k|$

Ainsi $\|T(x) - T(y)\|_1 \le (1-c)\,\|x - y\|_1$

avec **1 – c < 1**.

L'application T est donc **contractante**, d'où la convergence du processus itératif.

Partie II

Le projet « Doogle »

1) Principes et définitions

Afin d'appréhender plus précisément le fonctionnement de ce moteur de recherche, et la hiérarchisation des résultats dans les bases de données en général, **nous avons décidé de construire une réplique du célèbre moteur de recherches google**.

Fonctionnement originel de Google

A l'origine, le fonctionnement de Google se séparait en trois étapes qui ont chacune été grandement améliorées depuis :
- **L'indexation (deepcrawl),** durant laquelle le robot de Google (spider) parcourait le web et en aspirait une copie complète en cache. Cette phase est maintenant continue, et durait une dizaine de jours.
- **L'évaluation du Pagerank**, phase relativement secrète qui devait durer plusieurs jours.
- La création du classement, avec de nouveaux filtres sémantiques éventuels (la **dance**).

Pour des raisons techniques, nous avons décidé d'utiliser **un langage de programmation web, le PHP** (*Hypertext Preprocessor*). En effet, le moteur de recherche est conçu pour être utilisé par d'autres internautes sous la forme d'un site web, de façon analogue à Google. De plus, le PHP présente d'intéressantes fonctions d'édition de documents textes.

La mascotte de PHP

Créé en 1994 par Rasmus Lerdorf, ce langage de programmation est si riche qu'il est parfois considéré comme une plate-forme à part entière. Les scripts s'exécutent au niveau du serveur sur lequel se trouve la page web, et non sur l'ordinateur de l'utilisateur comme pour le Javascript, ce qui permet l'interaction avec une base de données.

C'est cette ultime particularité qui s'avèrera décisive pour le choix de notre programmation, comme nous le verrons dans la suite de la présentation. Le PHP est en effet conçu pour pouvoir être utilisé conjointement avec **les bases de données de type SQL**.

Le SQL (*Structured Query Language*) est un pseudo-langage informatique destiné à manipuler une base de donnée. Développée en 1974 par IBM, elle est aujourd'hui très largement répandue sur internet, en particulier grâce à MySQL, système de gestion des bases de données produit par Sun.

Base de données

Une base de données est une structure permettant le stockage d'informations, organisée et structurée pour en faciliter l'exploitation. Les bases de données proposées par MySQL sont de type **relationnelles**, où les données sont stockées dans de grands tableaux appelés **tables** que l'on peut éventuellement mettre en relation.
Concrètement, l'utilisateur définit différentes colonnes où viendront se ranger les valeurs stockées. Il est pratique de choisir une colonne “clé” qui servira de référence.

Considérons un exemple de base de données où sont accumulées des informations sur les élèves d'une classe. On peut imaginer la présence d'une première table **Eleves** définie de la façon suivante :

Eleves					
ID	Nom	Prénom	Date. Naissance	Téléphone	Adresse

Et qui donnerait, après insertion de quelques exemples aléatoires :

Eleves					
ID	Nom	Prénom	Date. Naissance	Téléphone	Adresse
1	Bernard	Elodie	14-07-1989	0?-??-??-??-??	12, rue...
2	Dubois	Julien	19-11-1989	0?-??-??-??-??	24, allée...
3	Martin	Laura	26-03-1989	0?-??-??-??-??	3, boulevard...
...					

La colonne ID (identifiant) est ici un exemple de colonne de référence : elle sert à repérer les élèves en leur attribuant un numéro, pour plus de facilité dans le tri des données. Par exemple, si l'on imagine une seconde table **Mathématiques** où seraient stockées les notes de mathématiques des élèves, il suffirait de les repérer par leurs identifiants.

Mathématiques					
ID	Interrogation 1	DS 1	Interrogation 2	DNS 1	DS 3
1	9	5	15	12	10
2	14	8	12	18	8
3	18	12	10	16	11
...					

On peut ainsi définir un nombre de table et de colonnes limité uniquement par les capacités techniques des machines utilisées. On peut également définir des valeurs par défauts pour les différentes colonnes (par exemple “Inconnu” pour la colonne adresse, valeur qui sera utilisée si on n'en précise aucune à l'insertion dans la base de données).

Pour ce projet, nous avons adopté le point de vue des cadres de Google lors de la mise au point de leur algorithme, afin de redécouvrir les secrets qu'ils gardent aujourd'hui précieusement. Nous avons vite compris que notre programme se déroulerait en trois étapes clé :

Programmes

- **Recensement de toutes les pages présentes sur le réseau.**

Cette opération sera effectuée par une page que nous avons appelé “Dooglebot”, nom choisi en hommage aux **ingénieurs de la firme de Mountain View qui ont appelé le leur Googlebot**. Il s'agira de parcourir chaque page pour y repérer les liens menant vers d'autres pages internet, afin de pouvoir calculer le pagerank par la suite. On repérera au passage l'adresse, le titre et les mots clés de la page. Dans le langage informatique, un tel robot est appelé **spider ou plus rarement crawler**.

- **Calcul du Pagerank de toutes les pages.**

Ceci sera permis par un algorithme nommé par analogie “Dooglerank”. Un tel algorithme est appelé **index software**. Il trie les résultats pour que le programme de recherche n'ait plus qu'à lire la liste déjà triée.

- **Interface du moteur de recherche à proprement parler.**

C'est la page internet sur laquelle vous arrivez en entrant http://www.google.fr dans votre navigateur internet. Nous l'avons simplement nommé “Doogle”. C'est le programme connu sous le nom de **query software** (engin de requête).

Intuitivement, nous avions décidé d'utiliser deux tables dans une base de données :

Tables dans la base de données

- **La table site**

Celle-ci recenserait la liste des pages web et stockerait les informations les concernant (titre, pagerank...).

- **La table liens**

Dans celle-ci seront enregistrés tous les liens entre les différentes pages.

C'est grâce à cette base de données que nous pourrons effectuer nos recherches. Ces tables répondront donc au schéma de la page suivante.

Table Sites

Elle servira à enregistrer toutes les informations relatives aux pages web.

SITES					
ID	url	titre	pagerank	keywords	scan
Identifiant du site pour le repérer dans la base de données.	Adresse de la page web.	Titre de la page	Le Pagerank de la page	Les mots clés que le créateur de la page a définis	Un booléen définissant si la page a déjà été scannée par notre robot ou non.

Il nous faut ajouter quelques précisions :
L'adresse de la page web se présente sur internet sous la forme *http://www.site.com/dossier/page.html*. Pour des raisons techniques, la plupart de nos tests seront effectuées en réseau local, et l'adresse pourra avoir une forme différente.
Le pagerank de la page sera calculé par un algorithme de notre fabrication.
Les mots clés sont présents sur la page dans des balises spéciales, mis à la disposition des robots de moteurs de recherche par les créateurs des sites internet.

Notre robot recenseur, comme évoqué plus haut, parcourra le web en analysant les pages une par une, afin de recenser tous les liens sortants des pages étudiées. A chaque lien trouvé, il ajoutera la nouvelle page dans la table **SITES**, avec une valeur de *scan=0*. Le robot analysera ensuite toutes les pages de **SITES** de valeur s*can=0*. Il mettra ainsi la liste à jour, recensant de nouveaux liens. Ceci lui permettra de parcourir l'intégralité du réseau. Ci-dessous, un court extrait de la table obtenue après parcours du web et calcul du pagerank, à titre d'exemple :

SITES					
ID	url	titre	pagerank	keywords	scan
1	.../www.polytech.unice.fr/index.html	Page d'accueil de j. leroux	0.15	Joël Leroux	1
2	.../www.polytech.unice.fr/coursprobas.html	Pas de titre	0.16		1
...					

Table Liens

Elle servira à enregistrer la liste des liens qui relient les sites entre eux. N'oublions pas que ceux-ci sont orientés.

Liens	
e	s
« entrée » du lien, identifiant de la page web d'où il provient.	« sortie » du lien, identifiant de la page web où il mène.

Le robot recenseur mettra à jour la liste des liens structurant le réseau. Ci-dessous, un court extrait de la table obtenue après parcours du web, à titre d'exemple :

Liens	
e	s
1	2
1	3
...	...
2	3
2	4
...	...
4	2

D'après ces bases, nous avons conçu un moteur de recherche à l'image de Google. Nous ne disposons pas des machines colossales de la firme de Mountain View, à la puissance de calcul exceptionnelle. Plutôt que d'utiliser une seule machine surpuissante, Google utilise en effet les principes du ***cloud computing*** : ils mettent en commun de nombreux ordinateurs. En effet, Google possède, d'après les estimations, entre **0.5 et 1 million de machines**, réparties entre 30 et 60 ***datacenter***, chacun consommant autant d'électricité qu'une petite ville (pour les machines et leur refroidissement).
Pour exploiter cette puissance phénoménale, ils ont conçu leur propre système de fichier, *GoogleFS*, qui met en commun l'espace disque de ces machines de façon optimale. Un algorithme nommé *Mapreduce* leur permet d'utiliser au mieux cet espace en triant près de 1000 go par 1000 machines en près d'une minute.
Ne disposant pas de telles technologies, nous avons donc décidé de ne travailler que sur une petite portion du web. A l'aide d'un logiciel spécialisé, **nous avons aspiré environ 700 pages web**. De là, nous avons conçu les programmes dont voici le fonctionnement schématique et les codes sources.

Le projet « Doogle »

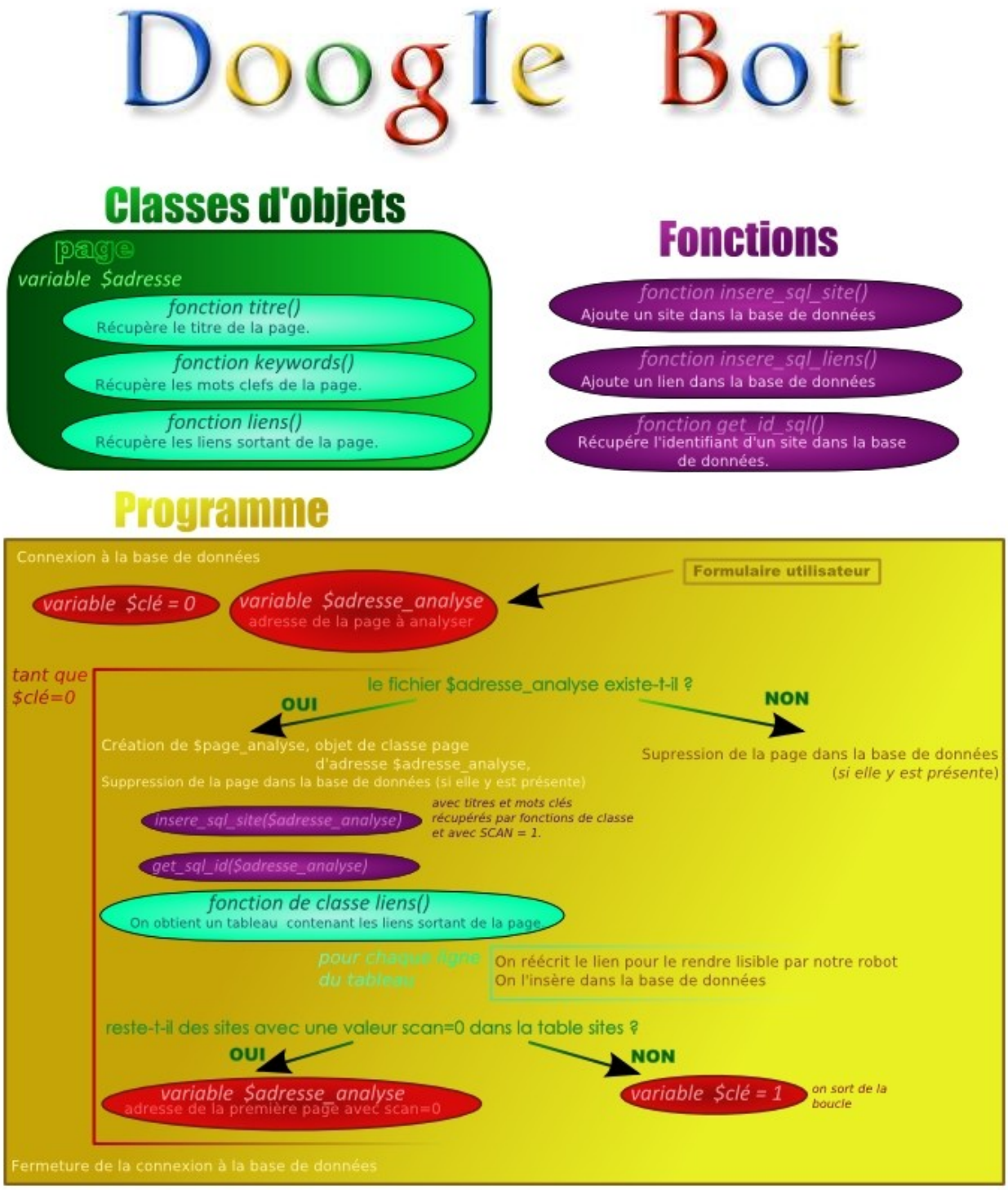
2) Codes source

Code source PHP : DoogleBot.php

```
<html>
<head><title>Doogle - Robot de recensement</title></head>
<body><?php

// Définition de classes

class page
{
 var $adresse;

  function page($adresse)  {
   $this->adresse = $adresse;
  }

  function titre()  {
   $fichier=file_get_contents($this->adresse) or die ("Page non existante");
   preg_match_all("#(?:<title>)(.+)(?:</title>)#isU", $fichier, $titre);
   return ($titre[1][0]);
  }

  function keywords()  {
   $fichier=file_get_contents($this->adresse) or die ("Page non existante");
   preg_match_all("#(?:keywords)(?:.+)(?:content=\")(.+)(?:\")#i", $fichier, $keywords);
   return ($keywords[1][0]);
  }

  function liens()  {
   $fichier=file_get_contents($this->adresse) or die ("Page non existante");
   preg_match_all("#(?:<a(?:.+)href=\")(.+)(?:\")#isU", $fichier, $liens);
   return ($liens[1]);
  }

}

 // Définition de fonctions

function insere_sql_site($url,$titre="Titre inconnu",$keywords="",$scan=0)
 {
$requete = mysql_query("SELECT * FROM sites WHERE url = '$url'") or die(mysql_error());
  $donnees = mysql_fetch_assoc($requete);
  if ( empty ($donnees)) {
     mysql_query("INSERT INTO sites (url,titre,keywords,scan)
                 VALUES  ('$url','$titre','$keywords','$scan')") or die(mysql_error());
  }
 }

function insere_sql_liens($e,$s)
 {
  $requete = mysql_query("SELECT * FROM liens WHERE e = '$e' AND s='$s'") or die(mysql_error());
  $donnees = mysql_fetch_assoc($requete);
```

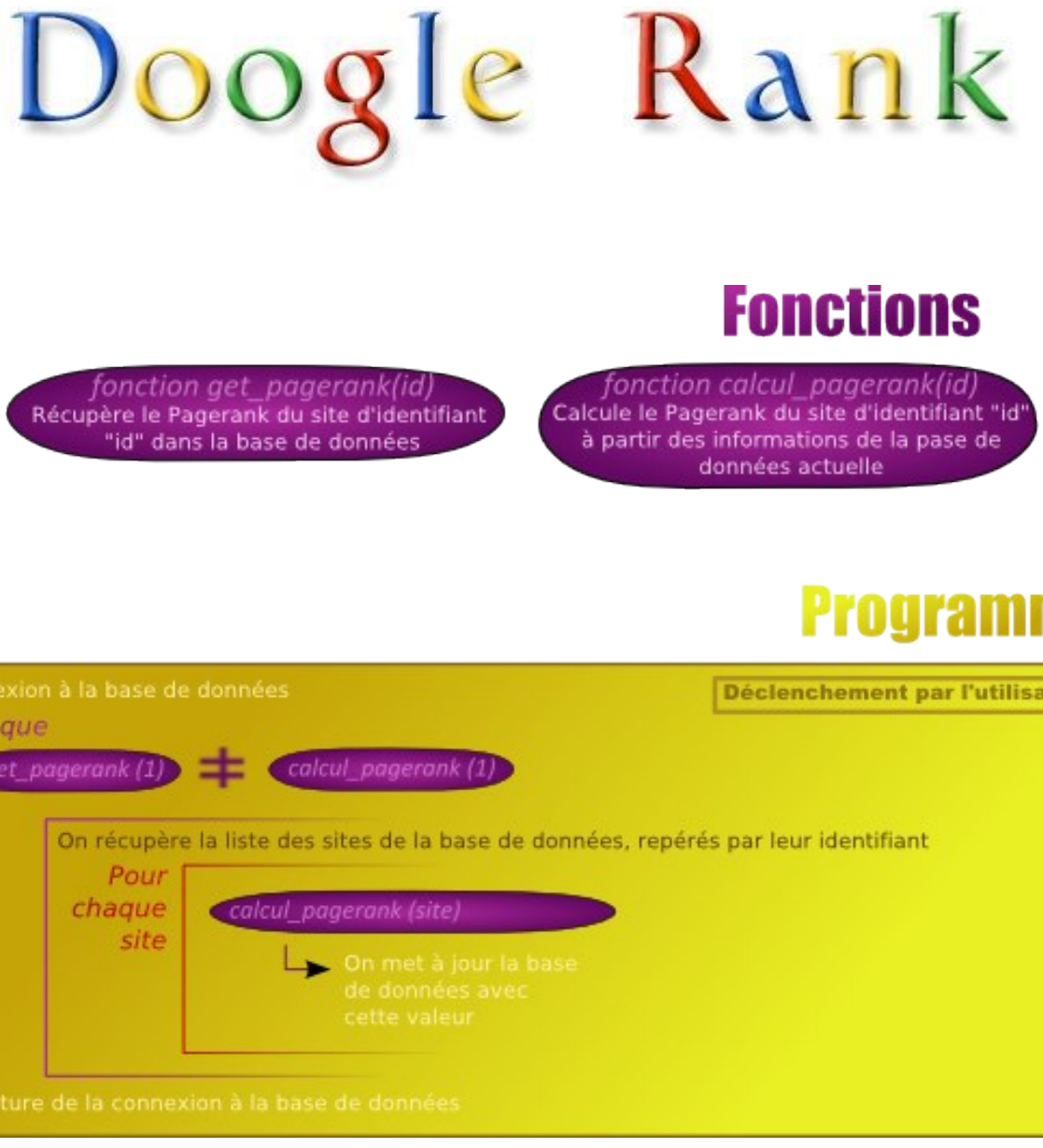
```
   if ( empty ($donnees)) {
      mysql_query("INSERT INTO liens (e,s) VALUES  ('$e','$s')") or die(mysql_error());
   }
  }
function get_sql_id($url)
 {
   $requete = mysql_query("SELECT id FROM sites WHERE url = '$url'") or die(mysql_error());
   $donnees = mysql_fetch_assoc($requete);
   return $donnees['id'];
 }

 // Code effectif

if ( isset($_GET['valider'])) {
$adresse_analyse = "E:/Programmes/wamp/www/Boogle";
$adresse_analyse .= $_GET['adresse'];
$connexion = mysql_connect('localhost', 'client', 'password');
mysql_select_db('boogle',$connexion);
$cle = 0;
while ($cle==0)
{

  if (! file_exists($adresse_analyse)) {
     mysql_query("DELETE FROM sites WHERE url = '$adresse_analyse'") or die(mysql_error());
     echo "Page " . $adresse_analyse . " non trouvée !<br />";
  }
  else {
     $page_analyse = new page($adresse_analyse);
     preg_match_all("#^(.*)/#i", $adresse_analyse, $racine);
     mysql_query("DELETE FROM sites WHERE url = '$adresse_analyse'") or die(mysql_error());
     insere_sql_site($adresse_analyse,$page_analyse->titre(),$page_analyse->keywords(),1);
     $id_analyse = get_sql_id($adresse_analyse);

     foreach ( $page_analyse->liens() as $url )
     {
       if ((! strstr($url, 'www')) && (! strstr($url, 'http')))
        {
          preg_match_all("#/?../(.*)$#i", $url, $urlt);
          while (isset ($urlt[1][0]) ) {
             $url=$urlt[1][0];
             preg_match_all("#^(.*)/#i", $racine[1][0], $racine);
             preg_match_all("#/?../(.*)$#i", $urlt[1][0], $urlt);
           }
          $url = $racine[1][0] . "/" . $url;
        }

       if (file_exists($url))
        {
          insere_sql_site($url);
          $id_lien = get_sql_id($url);
          insere_sql_liens($id_analyse,$id_lien);
        }
     }
     echo "Page " . $adresse_analyse . " analysée avec succès !<br />";
  }

   $requete = mysql_query("SELECT * FROM sites WHERE scan=0") or die(mysql_error());
   $a_scanner = mysql_fetch_assoc($requete);
```

```
   if ( empty ($a_scanner)) {
     $cle = 1;
   }
   else {
     $adresse_analyse = $a_scanner['url'];
   }
}
mysql_close($connexion);
}
?>

<br>Bienvenue sur le programme de recensement DoogleBot.<br>

<form action="booglebot.php" method="get">
  <input value="Adresse de la page de départ"  name="adresse" type="text">
  <input value="Lancer le programme" name="valider" type="submit">
</form>

</body></html>
```


Doogle Bot
Classes d'objets
page
variable $adresse
fonction titre()
Récupère le titre de la page.
fonction keywords()
Récupère les mots clefs de la page.
fonction liens()
Récupère les liens sortant de la page.
Fonctions
fonction insere_sql_site()
Ajoute un site dans la base de données
fonction insere_sql_liens()
Ajoute un lien dans la base de données
fonction get_id_sql()
Récupère l'identifiant d'un site dans la base de données.
Programme
Connexion à la base de données
Formulaire utilisateur
variable $clé = 0
variable $adresse_analyse
adresse de la page à analyser
tant que $clé=0
le fichier $adresse_analyse existe-t-il ?
OUI
NON
Création de $page_analyse, objet de classe page d'adresse $adresse_analyse,
Suppression de la page dans la base de données (si elle y est présente)
Supression de la page dans la base de données (si elle y est présente)
insere_sql_site($adresse_analyse)
avec titres et mots clés récupérés par fonctions de classe et avec SCAN = 1.
get_sql_id($adresse_analyse)
fonction de classe liens()
On obtient un tableau contenant les liens sortant de la page
pour chaque ligne du tableau
On réécrit le lien pour le rendre lisible par notre robot
On l'insère dans la base de données
reste-t-il des sites avec une valeur scan=0 dans la table sites ?
OUI
NON
variable $adresse_analyse
adresse de la première page avec scan=0
variable $clé = 1
on sort de la boucle
Fermeture de la connexion à la base de données

Code source PHP : DoogleRank.php

```
<html>
<head><title>Doogle - Calcul du Pagerank</title></head>
<body><?php

 // Définition de fonctions

function get_pagerank($id_page)
 {
  $requete = mysql_query("SELECT pagerank FROM sites WHERE id='$id_page'") or die(mysql_error());
  $donnees = mysql_fetch_assoc($requete);
  return $donnees['pagerank'];
 }

function calcul_pagerank($id_page)
 {
 $d = 0.85;
 $liens_entrants = mysql_query("SELECT * FROM liens WHERE s = '$id_page'") or die(mysql_error());
 $pagerank_page = 1 - $d;
 while ($lien  = mysql_fetch_assoc($liens_entrants)) {
    $source = $lien['e'];
    $pagerank_source = get_pagerank ($source);
    $requete = mysql_query("SELECT * FROM liens WHERE e = '$source'") or die(mysql_error());
    $compte = mysql_num_rows($requete);
    $pagerank_page += $d * $pagerank_source / $compte;
   }
 return $pagerank_page;
 }

 // Code effectif

if ( isset($_GET['valider']))
 {
$connexion = mysql_connect('localhost', 'client', 'password');
mysql_select_db('doogle',$connexion);

while (round(get_pagerank(1),2) <> round(calcul_pagerank(1),2))
 {
  $liste_site = mysql_query("SELECT * FROM sites ORDER BY id ASC") or die(mysql_error());
  while($site = mysql_fetch_assoc($liste_site))  {
     $id = $site['id'];
     $nouveau_pagerank = calcul_pagerank ( $id );
     mysql_query("
                 UPDATE sites SET pagerank=$nouveau_pagerank WHERE id='$id'
                 ") or die(mysql_error());
    }
  echo "Tour de mise à jour effectué avec succès.<br /><br />";
 }
mysql_close($connexion);
  }

?>

<br>Bienvenue sur le programme de calcul DoogleRank.<br>
<form action="dooglerank.php" method="get">
<input value="Démarer le calcul" name="valider" type="submit">
</form>

</body></html>
```


Doogle Rank
Fonctions
fonction get_pagerank(id)
Récupère le Pagerank du site d'identifiant "id" dans la base de données
fonction calcul_pagerank(id)
Calcule le Pagerank du site d'identifiant "id" à partir des informations de la pase de données actuelle
Programme
Connexion à la base de données
Déclenchement par l'utilisateur
tant que
get_pagerank (1)
≠
calcul_pagerank (1)
On récupère la liste des sites de la base de données, repérés par leur identifiant
Pour chaque site
calcul_pagerank (site)
On met à jour la base de données avec cette valeur
Fermeture de la connexion à la base de données

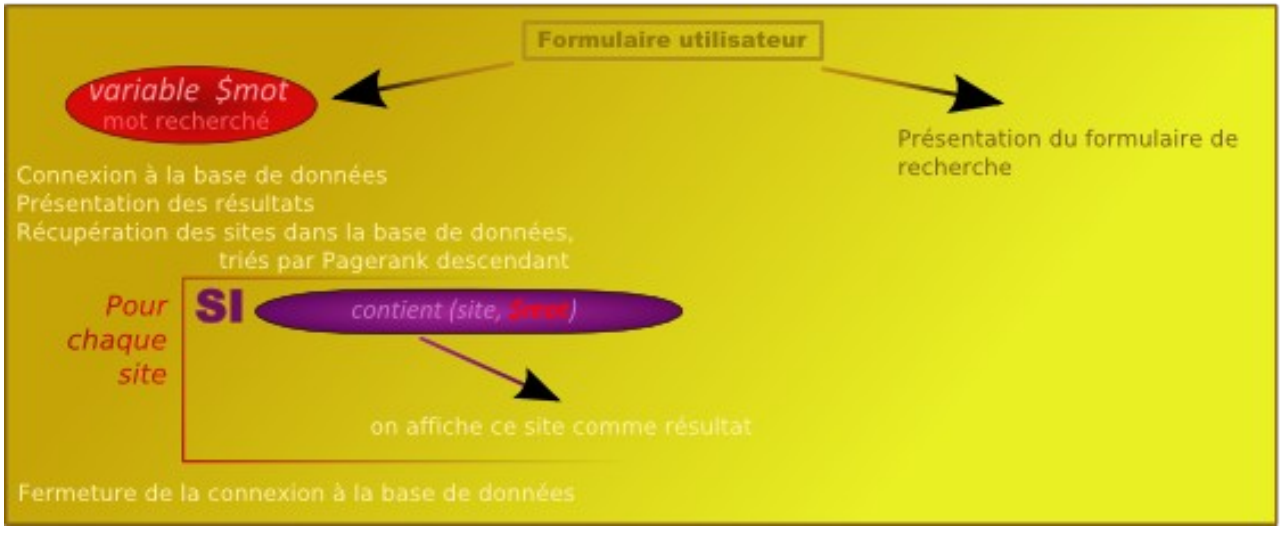
CODE SOURCE PHP : DOOGLE.PHP

```
<html>
<head><title>Doogle - Module de recherche</title></head>

<body><?php


 // Définition de fonctions


function contient($terme,$adresse)  {
 $fichier=file_get_contents($adresse) or die ("Page non existante");
 preg_match_all("#(.){10}($terme)(.){0,100}#i", $fichier, $resultat);
 if (isset ($resultat[0][0])) {
   return true;
 }
 else {
   return false;
 }
}




 // Code effectif


if ( isset($_GET['recherche']))
{
$mot = $_GET['recherche'];
?>
<br />
<table width="100%" border=0><tr>
<td width=200><img src="doogle.jpg" width="200"></td>
<td> <form action="doogle.php" method="get">
  <input value="<?php echo $mot; ?>" name="recherche" type="text" size="55" >
<input value="Recherche Doogle" name="valider" type="submit">
     </form></td>
</tr></table>
<br /><br />

<?php
$connexion = mysql_connect('localhost', 'client', 'password');
mysql_select_db('doogle',$connexion);

$requete = mysql_query("SELECT * FROM sites ORDER BY pagerank DESC") or die(mysql_error());
while($site = mysql_fetch_row($requete))
  {
   $url = $site[0];
   if (contient($_GET['recherche'],$url)) {
      if (empty ($site[1])) { $titre = $url; }
      else {$titre = $site[1]; }
      echo "<a href=\"file:///" . $url . "\">" . $titre . "</a><br />";
   }
  }
mysql_close($connexion);
}
```

```
else {
 ?>
 <br /><br /><br />
 <center><img src="doogle.jpg"><br /><br />
 <form action="doogle.php" method="get">
 <input name="recherche" type="text" size="55" ><br />
 <input value="Recherche Doogle" name="valider" type="submit">
 </form>
 </center>
 <br />
 <?php
}
?>

</body></html>
```

Fonctions

Programme



Le projet « Doogle »

3) Mise en place et résultats

Pour des raisons techniques évidentes, nous ne pouvions pas nous permettre d'analyser, comme le fait Google, les 1000 milliards de pages présentes sur le web. Nous avons utilisé un logiciel appelé **aspirateur de sites** pour télécharger une partie d'internet sur notre PC, où nous effectuerons les tests. En effet, si nous lancions notre robot de recensement sur internet, il ne s'arrêterait qu'après avoir référencé tous les sites, ce qui prendrait un temps fou et surchargerait la mémoire de notre matériel.

Nous avons choisi au hasard dans la liste des résultats de Google sur les transformées de Fourier un site, qui s'avéra être le site web de Joel Leroux, sur le nom de domaine de l'école polytechnique de Nice, qui proposait un exposé très complet sur les transformées de Fourier et autres outils informatiques. Nous n'avons utilisé pour ces démonstrations qu'**une infime partie de ce site (703 pages)**, ce qui donnera des **résultats bien évidemment grandement approximés**. Cependant, nous pourrons toujours les utiliser pour vérifier la cohérence des résultats.

Toutes les recherches témoignent d'une même structure des résultats : en premier, elles offrent des pages très générales de sommaires, qui donnent à l'internaute une vue d'ensemble du site qui mentionne le terme qui l'intéresse. Dans un second temps, d'autres sommaires plus spécialisés permettent un accès à une partie plus spécifique mais plus restreinte. Finalement arrivent les pages, dans l'ordre de pertinence, ou plus exactement de popularité. Les premières sont généralement les plus claires et les plus complètes, ce qui est du au fait qu'elles sont de nombreuses fois citées.

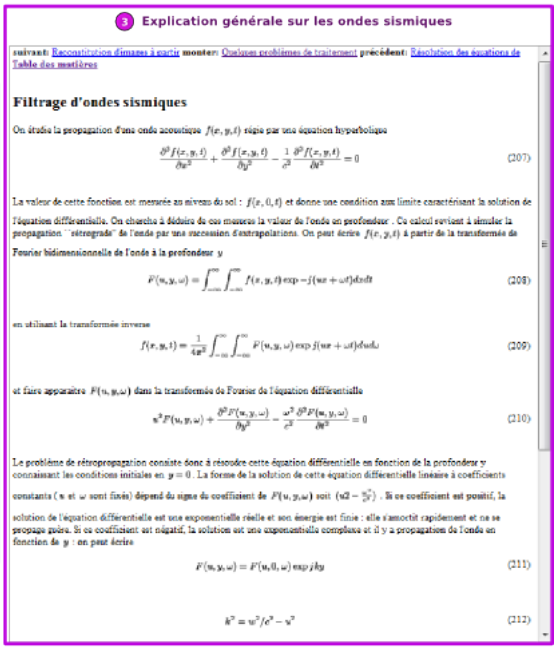
Doogle ondes Recherche Doogle


1 Sommaire général du site

Table des matières


2 Sommaire plus précis (sous-partie)

Quelques problèmes de traitement de signaux multidimensionnels


3 Explication générale sur les ondes sismiques

Filtrage d'ondes sismiques

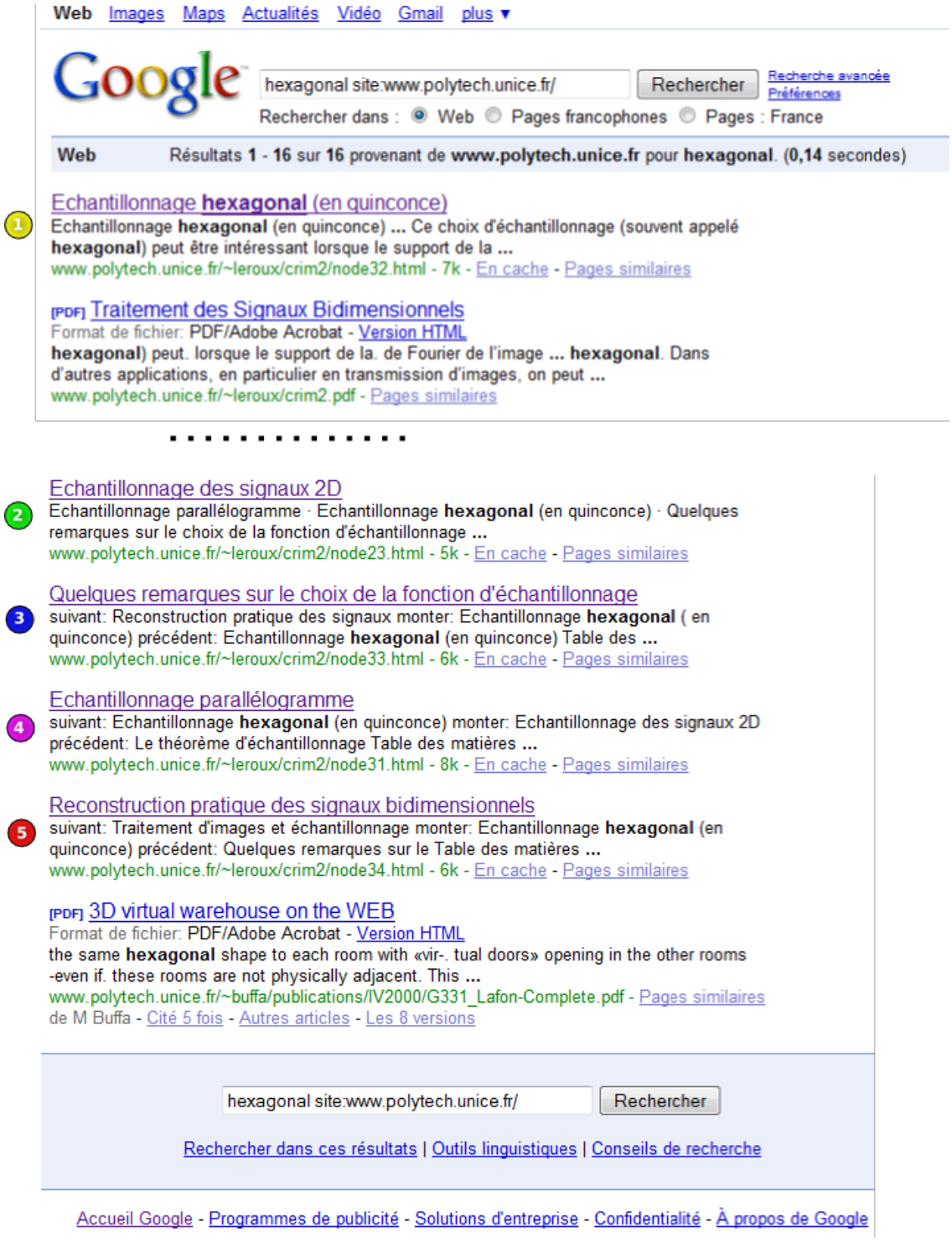
Comme nous allons le présenter dans les pages suivantes, les résultats des recherches, bien que concluants, montrent quelques divergences avec Google. Certaines recherches donnent des résultats identiques sur Google et Doogle, comme par exemple le mot « racines ». Mais parfois, l'ordre ne se retrouve que partiellement, voire bouleversé. Ceci peut néanmoins s'expliquer facilement.

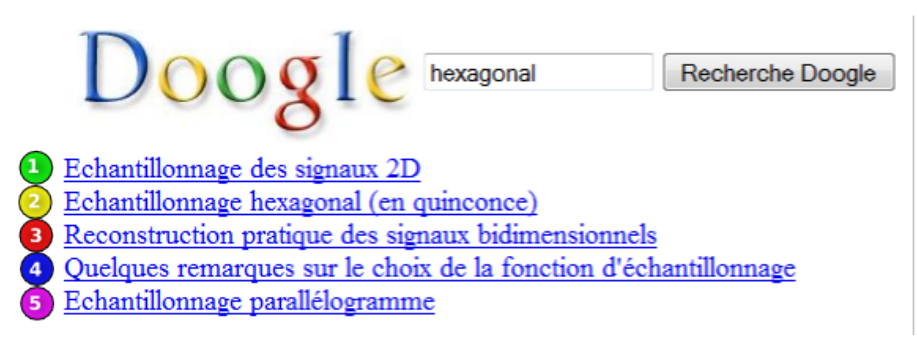
En effet, nous n'avons aspiré qu'une infime partie du web, et le calcul des pagerank s'en retrouve donc extrêmement lacunaire, puisqu'il dépend en grande partie du nombre de liens pointant vers la page. Nous avons pu ignorer des milliers de sites qui pointaient vers les pages étudiées, ce qui déséquilibre leur Pagerank. Pour des comparaisons plus fiables, il faut bien entendu augmenter le nombre de sites étudiés.

D'autre part, il est à signaler que la firme de Mountain View a ajouté de nombreux filtres sémantiques et des programmes espions pour cerner le comportement de l'utilisateur et répondre au mieux à sa demande, en se basant par exemple sur son historique de recherche.

Il faut finalement rappeler que, puisque tout internet n'a pas été aspiré, la liste de Google est bien évidemment beaucoup plus fournie que celle de notre programme. C'est pour cela qu'à certains endroits, nous nous sommes permis d'occulter certains passages des résultats de Google qui ne contenaient pas de sites aspirés. Par soucis de clarification, nous avons toutefois réduit l'intervalle de nos recherches Google au nom de domaine des sites aspirés grâce à la commande de syntaxe Google :
« site:www.polytech.unice.fr/ »

1. Interprétation en termes de lieu des racines
2. Stabilité des filtres causaux
3. Filtrage des signaux bidimensionnels


Web Images Maps Actualités Vidéo Gmail plus ▾
Google
hexagonal site:www.polytech.unice.fr/
Rechercher
Recherche avancée
Préférences
Rechercher dans : Web Pages francophones Pages : France
Web Résultats 1 - 16 sur 16 provenant de www.polytech.unice.fr pour hexagonal. (0,14 secondes)
1 Echantillonnage hexagonal (en quinconce)
Echantillonnage hexagonal (en quinconce) ... Ce choix d'échantillonnage (souvent appelé hexagonal) peut être intéressant lorsque le support de la ...
www.polytech.unice.fr/~leroux/crim2/node32.html - 7k - En cache - Pages similaires
[PDF] Traitement des Signaux Bidimensionnels
Format de fichier: PDF/Adobe Acrobat - Version HTML
hexagonal) peut. lorsque le support de la. de Fourier de l'image ... hexagonal. Dans d'autres applications, en particulier en transmission d'images, on peut ...
www.polytech.unice.fr/~leroux/crim2.pdf - Pages similaires
2 Echantillonnage des signaux 2D
Echantillonnage parallélogramme · Echantillonnage hexagonal (en quinconce) · Quelques remarques sur le choix de la fonction d'échantillonnage ...
www.polytech.unice.fr/~leroux/crim2/node23.html - 5k - En cache - Pages similaires
3 Quelques remarques sur le choix de la fonction d'échantillonnage
suivant: Reconstruction pratique des signaux monter: Echantillonnage hexagonal (en quinconce) précédent: Echantillonnage hexagonal (en quinconce) Table des ...
www.polytech.unice.fr/~leroux/crim2/node33.html - 6k - En cache - Pages similaires
4 Echantillonnage parallélogramme
suivant: Echantillonnage hexagonal (en quinconce) monter: Echantillonnage des signaux 2D précédent: Le théorème d'échantillonnage Table des matières ...
www.polytech.unice.fr/~leroux/crim2/node31.html - 8k - En cache - Pages similaires
5 Reconstruction pratique des signaux bidimensionnels
suivant: Traitement d'images et échantillonnage monter: Echantillonnage hexagonal (en quinconce) précédent: Quelques remarques sur le Table des matières ...
www.polytech.unice.fr/~leroux/crim2/node34.html - 6k - En cache - Pages similaires
[PDF] 3D virtual warehouse on the WEB
Format de fichier: PDF/Adobe Acrobat - Version HTML
the same hexagonal shape to each room with «vir-. tual doors» opening in the other rooms -even if. these rooms are not physically adjacent. This ...
www.polytech.unice.fr/~buffa/publications/IV2000/G331_Lafon-Complete.pdf - Pages similaires
de M Buffa - Cité 5 fois - Autres articles - Les 8 versions
hexagonal site:www.polytech.unice.fr/
Rechercher
Rechercher dans ces résultats | Outils linguistiques | Conseils de recherche
Accueil Google - Programmes de publicité - Solutions d'entreprise - Confidentialité - À propos de Google


Doogle
hexagonal
Recherche Doogle
1 Echantillonnage des signaux 2D
2 Echantillonnage hexagonal (en quinconce)
3 Reconstruction pratique des signaux bidimensionnels
4 Quelques remarques sur le choix de la fonction d'échantillonnage
5 Echantillonnage parallélogramme

Pour la recherche du terme « propagation », on voit de même que malgré certaines divergences évidentes, l'ordre partiel est conservé.

Le projet « Doogle »

4) Améliorations

Afin d'optimiser nos résultats, nous avons décidé d'adopter une nouvelle méthode de recherche. Au lieu de parcourir l'ensemble des pages à chaque requête, nous avons entrepris d'utiliser une table annexe nommée **Mots**, où nous enregistrerons la totalité des mots de longueur significative (pour éviter les mots comme « et », « de », « pour » trop banals) présents sur les sites. C'est dans cette base de données que nous effectuerons ensuite notre recherche. Il faudra pour cela modifier l'algorithme du robot recenseur, afin qu'il remplisse cette nouvelle table.

Table Mots

Elle recensera tous les mots de plus de 4 lettres sur les pages web.

Mots	
mot	site
Le mot en lui-même, une chaine de caractères	L'identifiant de la page où ce mot a été aperçu

Ci-dessous, un court extrait de la table obtenue après parcours du web, à titre d'exemple :

Liens	
mot	site
...	...
traitement	24
numérique	24
signaux	24
objectifs	24
cours	24
...	...

Nous n'aurons ainsi plus à considérer les nombreuses pages ne contenant pas le terme recherché. Tous les mots seront référencés une fois pour toutes. Le temps d'exécution devrait être beaucoup plus rapide, mais le temps de recensement devrait être plus long. En effet, les fonctions de parsage (décomposition de document) sont sensiblement plus lourdes que les requêtes SQL, d'autant plus qu'elles seront beaucoup moins nombreuses ici. Le temps d'exécution d'une recherche sera un simple parcours de la base de données, optimisée à cet effet.

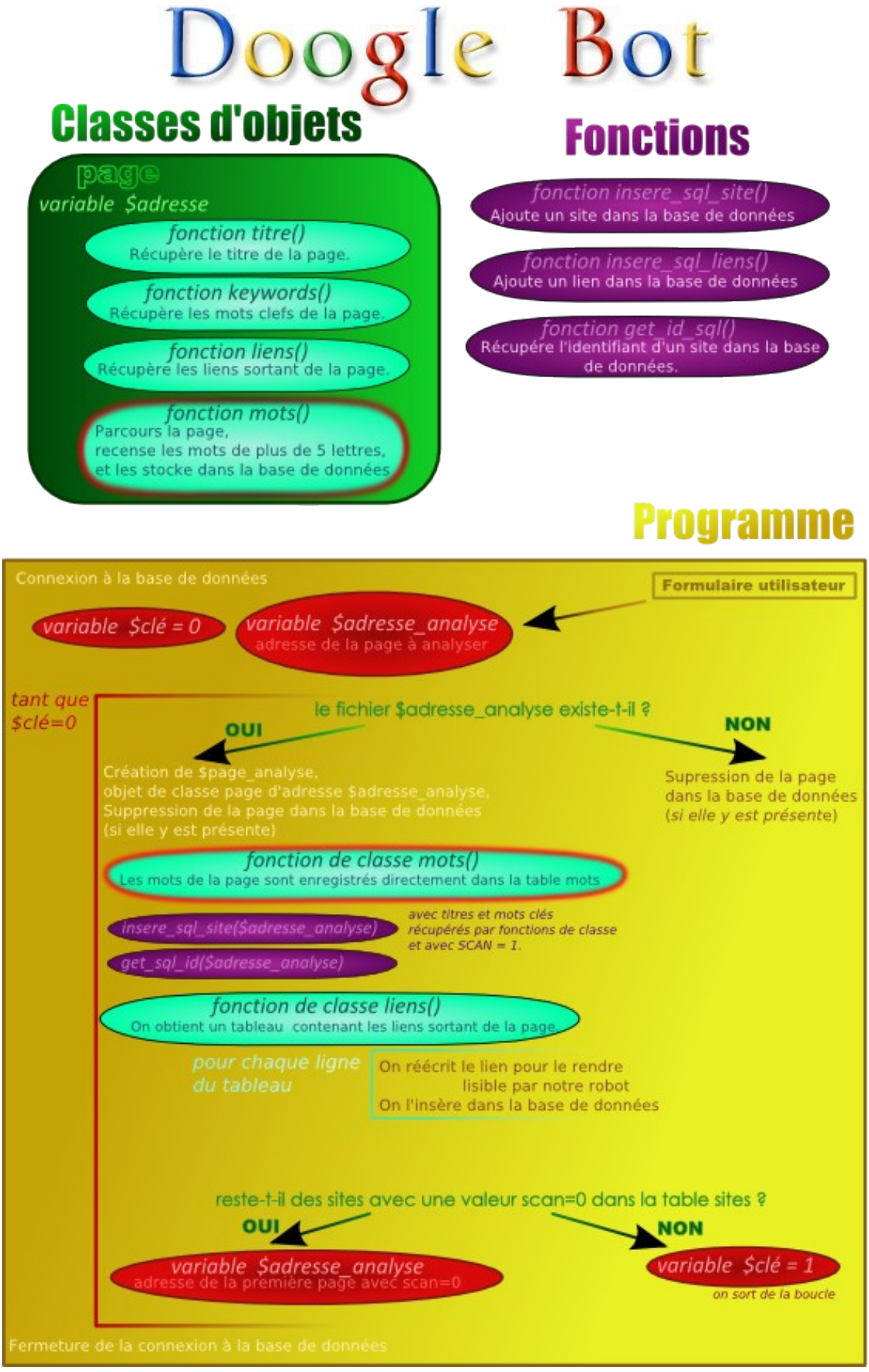
Vous trouverez à la prochaine page le code source des pages modifiées :

Code source PHP : DoogleBot.php

```
<html>
<head><title>Doogle - Robot de recensement</title></head>
<body><?php

// Définition de classes

class page
{
 var $adresse;
  function page($adresse)  {
   $this->adresse = $adresse;
  }
  function titre()  {
   $fichier=file_get_contents($this->adresse) or die ("Page non existante");
   preg_match_all("#(?:<title>)(.+)(?:</title>)#isU", $fichier, $titre);
   return ($titre[1][0]);
  }
  function keywords()  {
   $fichier=file_get_contents($this->adresse) or die ("Page non existante");
   preg_match_all("#(?:keywords)(?:.+)(?:content=\")(.+)(?:\")#i", $fichier, $keywords);
   return ($keywords[1][0]);
  }
  function liens()  {
   $fichier=file_get_contents($this->adresse) or die ("Page non existante");
   preg_match_all("#(?:<a(?:.+)href=\")(.+)(?:\")#isU", $fichier, $liens);
   return ($liens[1]);
  }
  function insere_mots($id) {
   $fichier=file_get_contents($this->adresse) or die ("Page non existante");
   preg_match_all("|<[^>]+>(.*)</[^>]+>|iU", $fichier, $phrases);
   if (isset($phrases[1][0])) {
     $phrases = $phrases[1];
   }
   else {
     $phrases = array();
     $phrases[0] = $fichier;
   }

   foreach ($phrases as $phrase)
   {
    preg_match_all("#([\w]{5,15}):?( |,|;|.|:|'|"\$)#iU", $phrase, $mots);

    foreach ($mots[1] as $mot)
     {
      $mot = strtolower($mot);
      $query = mysql_query("SELECT * FROM mots WHERE mot='$mot' AND
      sites='$id'");
      if (mysql_num_rows($query) == 0) {
      mysql_query("INSERT INTO mots (mot,sites) VALUES ('$mot','$id')") or
      die(mysql_error());
       }
     }
   }
  }
}
```

```
// Définition de fonctions

function insere_sql_site($url,$titre="Titre inconnu",$keywords="",$scan=0)
 {
$requete = mysql_query("SELECT * FROM sites WHERE url = '$url'") or die(mysql_error());
  $donnees = mysql_fetch_assoc($requete);
  if ( empty ($donnees)) {
     mysql_query("INSERT INTO sites (url,titre,keywords,scan)
                 VALUES  ('$url','$titre','$keywords','$scan')") or die(mysql_error());
  }
 }

function insere_sql_liens($e,$s)
 {
  $requete = mysql_query("SELECT * FROM liens WHERE e = '$e' AND s='$s'") or die(mysql_error());
  $donnees = mysql_fetch_assoc($requete);
  if ( empty ($donnees)) {
     mysql_query("INSERT INTO liens (e,s) VALUES  ('$e','$s')") or die(mysql_error());
  }
 }

function get_sql_id($url)
 {
   $requete = mysql_query("SELECT id FROM sites WHERE url = '$url'") or die(mysql_error());
   $donnees = mysql_fetch_assoc($requete);
   return $donnees['id'];
 }

// Code effectif

if ( isset($_GET['valider'])) {
$adresse_analyse = "E:/Programmes/wamp/www/Doogle";
$adresse_analyse .= $_GET['adresse'];
$connexion = mysql_connect('localhost', 'client', 'password');
mysql_select_db('boogle',$connexion);
$cle = 0;
while ($cle==0)
{

  if (! file_exists($adresse_analyse)) {
     mysql_query("DELETE FROM sites WHERE url = '$adresse_analyse'") or die(mysql_error());
     echo "Page " . $adresse_analyse . " non trouvée !<br />";
  }
  else {
     $page_analyse = new page($adresse_analyse);
     preg_match_all("#^(.*)/#i", $adresse_analyse, $racine);
     mysql_query("DELETE FROM sites WHERE url = '$adresse_analyse'") or die(mysql_error());
     insere_sql_site($adresse_analyse,$page_analyse->titre(),$page_analyse->keywords(),1);
     $id_analyse = get_sql_id($adresse_analyse);
     $page_analyse->insere_mots($id_analyse);
```

```
    foreach ( $page_analyse->liens() as $url ) {
      if ((! strstr($url, 'www')) && (! strstr($url, 'http')))
       {
        preg_match_all("#/?../(.*)$#i", $url, $urlt);
        while (isset ($urlt[1][0]) ) {
           $url=$urlt[1][0];
           preg_match_all("#^(.*)/#i", $racine[1][0], $racine);
           preg_match_all("#/?../(.*)$#i", $urlt[1][0], $urlt);
         }
        $url = $racine[1][0] . "/" . $url;
       }

      if (file_exists($url))
       {
        insere_sql_site($url);
        $id_lien = get_sql_id($url);
        insere_sql_liens($id_analyse,$id_lien);
       }
    }
    echo "Page " . $adresse_analyse . " analysée avec succès !<br />";
   }

   $requete = mysql_query("SELECT * FROM sites WHERE scan=0") or die(mysql_error());
   $a_scanner = mysql_fetch_assoc($requete);
   if ( empty ($a_scanner)) {
     $cle = 1;
   }
   else {
     $adresse_analyse = $a_scanner['url'];
   }
}
mysql_close($connexion);
}
?>

<br>Bienvenue sur le programme de recensement DoogleBot.<br>

<form action="booglebot.php" method="get">
  <input value="Adresse de la page de départ"  name="adresse" type="text">
  <input value="Lancer le programme" name="valider" type="submit">
</form>

</body></html>
```


Doogle Bot
Classes d'objets
page
variable $adresse
fonction titre()
Récupère le titre de la page.
fonction keywords()
Récupère les mots clefs de la page.
fonction liens()
Récupère les liens sortant de la page.
fonction mots()
Parcours la page,
recense les mots de plus de 5 lettres,
et les stocke dans la base de données
Fonctions
fonction insere_sql_site()
Ajoute un site dans la base de données
fonction insere_sql_liens()
Ajoute un lien dans la base de données
fonction get_id_sql()
Récupère l'identifiant d'un site dans la base de données.
Programme
Connexion à la base de données
Formulaire utilisateur
variable $clé = 0
variable $adresse_analyse
adresse de la page à analyser
tant que $clé=0
le fichier $adresse_analyse existe-t-il ?
OUI
NON
Création de $page_analyse,
objet de classe page d'adresse $adresse_analyse,
Suppression de la page dans la base de données
(si elle y est présente)
Supression de la page dans la base de données
(si elle y est présente)
fonction de classe mots()
Les mots de la page sont enregistrés directement dans la table mots
insere_sql_site($adresse_analyse)
get_sql_id($adresse_analyse)
avec titres et mots clés récupérés par fonctions de classe et avec SCAN = 1.
fonction de classe liens()
On obtient un tableau contenant les liens sortant de la page
pour chaque ligne du tableau
On réécrit le lien pour le rendre lisible par notre robot
On l'insère dans la base de données
reste-t-il des sites avec une valeur scan=0 dans la table sites ?
OUI
NON
variable $adresse_analyse
adresse de la première page avec scan=0
variable $clé = 1
on sort de la boucle
Fermeture de la connexion à la base de données

Code source PHP : Doogle.php

```
<html>
<head><title>Doogle - Module de recherche</title></head>

<body><?php

if ( isset($_GET['recherche']))
{
$mot = $_GET['recherche'];
?>
<br />
<table width="100%" border=0><tr>
<td width=200><img src="doogle.jpg" width="200"></td>
<td> <form action="doogle.php" method="get">
  <input value="<?php echo $mot; ?>" name="recherche" type="text" size="55" >
<input value="Recherche Doogle" name="valider" type="submit">
     </form></td>
</tr></table>
<br /><br />

<?php
$connexion = mysql_connect('localhost', 'client', 'password');
mysql_select_db('doogle',$connexion);

$requete = mysql_query("SELECT url,titre FROM sites S,mots M WHERE
S.id=M.sites AND M.mot='".$mot."' ORDER BY S.pagerank DESC") or
die(mysql_error());
while($site = mysql_fetch_row($requete))
  {
   $url = $site[0];
   if (empty ($site[1])) { $titre = $url; }
   else {$titre = $site[1]; }
   echo "<a href=\"file:///" . $url . "\">" . $titre . "</a><br />";
  }
mysql_close($connexion);
}

else {
?>
<br /><br /><br />
<center><img src="doogle.jpg"><br /><br />
<form action="doogle.php" method="get">
<input name="recherche" type="text" size="55" ><br />
<input value="Recherche Doogle" name="valider" type="submit">
</form>
</center>
<br />
<?php
}
?>

</body></html>
```

Doogle Search

Programme

Nous avons effectué certains tests pour comparer les deux algorithmes.

Référencement des sites : DoogleBot.php	
Ancienne version (pas de table MOTS)	**Nouvelle version (table MOTS)**
Analyse d'un groupe de 100 pages : 32 secondes Analyse d'un groupe de 500 pages : 1 minute 44 secondes Analyse des 703 pages étudiées : 2 minutes 23 secondes **Temps moyen d'analyse pour une page :** **0,203 secondes**	Analyse d'un groupe de 100 pages : 2 minutes 19 secondes Analyse d'un groupe de 500 pages : 5 minutes 59 secondes Analyse des 703 pages étudiées : 7 minutes 04 secondes **Temps moyen d'analyse pour une page :** **0,603 secondes**
Module de recherches : Doogle.php	
Ancienne version (pas de table MOTS)	**Nouvelle version (table MOTS)**
Recherche de «fourier» (180 résultats) 50 secondes Recherche de «numérique» (73 résultats) 47 secondes Recherche de «huffman» (10 résultats) 41 secondes **Recherche de «télécommunications» (0 résultats)** **43 secondes** **Temps moyen pour un résultat :** **0,688 secondes**	Recherche de «fourier» (180 résultats) 0 secondes mesurées Recherche de «numérique» (73 résultats) 0 secondes mesurées Recherche de «huffman» (10 résultats) 0 secondes mesurées **Recherche de «télécommunications» (0 résultats)** **0 secondes mesurées** **Temps moyen pour un résultat :** **quasi instantané**

Dans l'ancienne version, le temps d'exécution d'une recherche était significativement plus long, ce qui est très gênant pour l'ergonomie de l'utilisateur, mais il ne dépendait surtout que du nombre de sites étudiés, si bien qu'une recherche sans résultats était aussi longue que n'importe quelle autre. La nouvelle version corrige ces défauts, et rend l'exécution quasiment instantanée.

Une recherche approfondie montre que Google utilise très probablement un système analogue : ses recherches sont effectuées dans une titanesque base de données. Il possède en effet une copie entière du World Wide Web sur ses machines, que l'internaute peut consulter sur son navigateur grâce à l'option « *visiter en cache* ». Ceci permet au moteur de recherche de proposer, en plus de son atout de pertinence, une **incroyable rapidité dans ses recherches** (quelques dixièmes de secondes tout au plus), par rapport à ses concurrents.

D'autre part, nous avons voulu améliorer le temps de calcul du Pagerank en limitant les recherches dans la base de données, qui y sont très nombreuses. Nous les avons réduites au strict nécessaire : la table de la base de données est enregistrée dans le script sous la forme d'un tableau, et c'est sur ce tableau que nous effectuons les calculs. Il n'y a ainsi plus que deux requêtes : une au début du programme pour récupérer le contenu de la table, et une à la fin du programme pour mettre à jour la table.

Tableau $liste_liens

C'est l'analogue de la table Liens de la base de données. Il est construit de la façon suivante :

Index de la case	Case
L'identifiant d'un site	Un tableau contenant les identifiants de tous les sites ayant un lien menant vers le site en index.

Chaque case est ici en effet repérée par un index, entier naturel. Par exemple, la table de gauche donnera le tableau de droite :

Liens	
e	s
1	2
1	3
...	...
2	3
2	4
...	...
4	2

$liste_liens	
Index	Case
1	Tableau : [2;3;....]
2	Tableau : [3;4;...]
3	Tableau : []
4	Tableau : [2;...]

Tableau $liste_sites

Mais le PHP offre la possibilité d'utiliser des tableaux multi-dimentionnels. Ceux-ci ne sont plus alors une succession de cases repérées par un index numérique, mais une table analogue à celle de la base de données dont les colonnes sont repérées par des mots clés. Le tableau analogue de la table **SITES** de la base de données sera construit de la façon suivante :

<table>
<tr><th>Index de la case</th><th>Case</th></tr>
<tr><td>L'identifiant d'un site</td><td>Un tableau multidimensionnel contenant les informations de la base de données sur ce site :
<table>
<tr><th>ID</th><th>url</th><th>titre</th><th>pagerank</th><th>keywords</th><th>scan</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
</table>

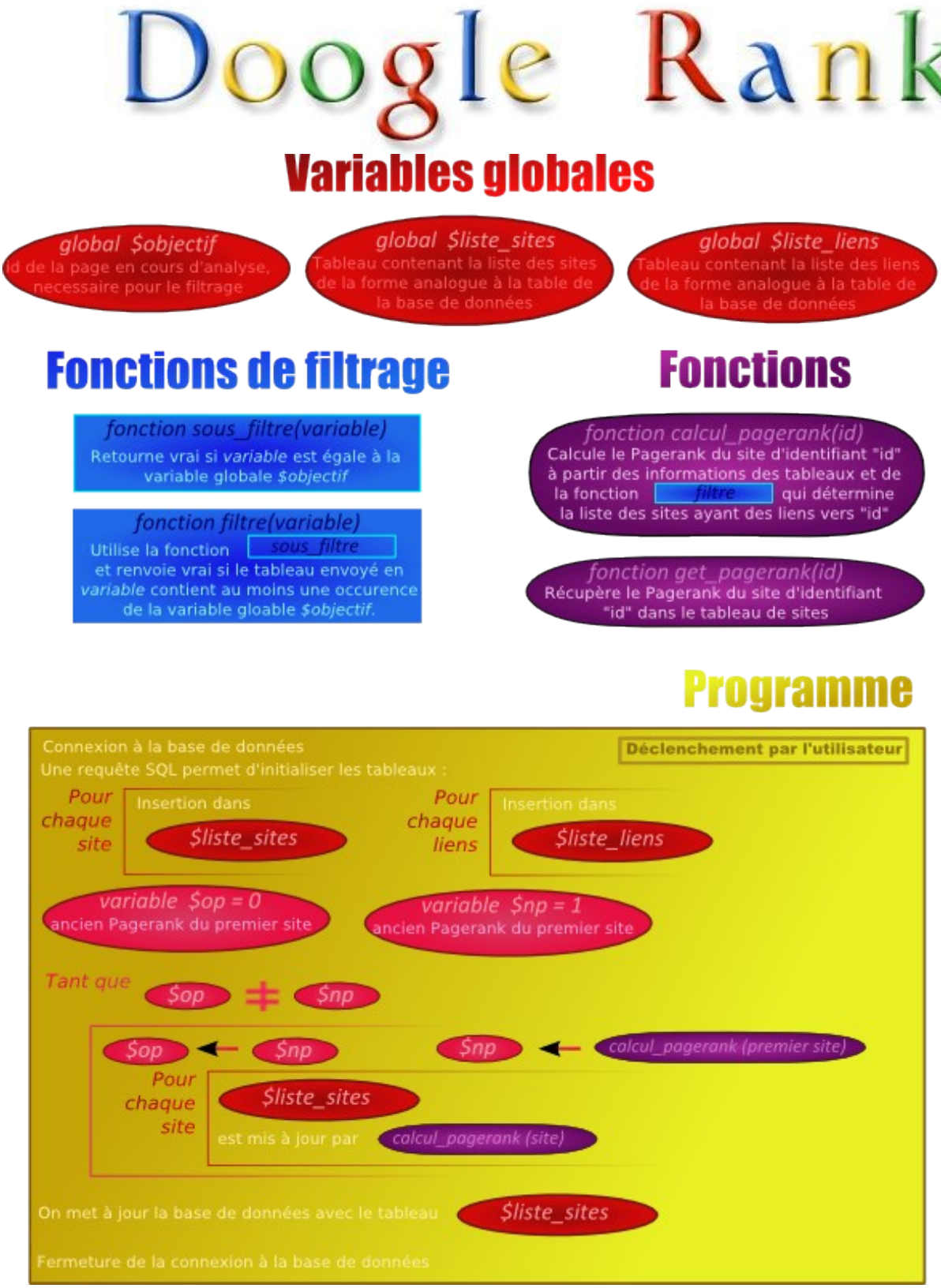
La structure restera donc analogue à celle de la base de données.

Le nouveau code obtenu est explicité dans les pages suivantes. Il se montrera, comme nous le verrons, plus efficace, mais il faut reconnaître qu'il est cependant bien moins clair.

Code source PHP : DoogleRank.php

```
<html>
<head><title>Doogle - Calcul du Pagerank</title></head>
<body><?php



 // Définition de fonctions de filtrage



function sousfiltre ($variable)
 {
  global $objectif;
  return ($variable == $objectif);
 }
function filtre($variable)
 {
  $tableau = (array_filter($variable, "sousfiltre"));
  return (isset ($tableau[0]));
 }




 // Définition de fonctions



function get_pagerank($id_page)
 {
 global $liste_sites;
 return $liste_sites[$id_page]['pagerank'];
 }
function calcul_pagerank($id_page)
 {
 global $liste_sites;
 global $liste_liens;
 $liste_liens_entrants = (array_filter($liste_liens, "filtre"));

 $d = 0.85;
 $pagerank_page = 1 - $d;
 foreach (array_keys ($liste_liens_entrants) as $source)
   {
   $pagerank_source = get_pagerank ($source);
   $compte = count($liste_liens_entrants[$source]);
   $pagerank_page += $d * $pagerank_source / $compte;
   }
 return $pagerank_page;
 }
```

```
 // Code effectif

if ( isset($_GET['valider']))
 {
$connexion = mysql_connect('localhost', 'client', 'password');
mysql_select_db('doogle',$connexion);

$requete = mysql_query("SELECT * FROM sites ORDER BY id ASC") or die(mysql_error());
$liste_sites = array();

while ( $donnees = mysql_fetch_assoc($requete))  {
  $liste_sites[$donnees['id']] = $donnees;
 }

$requete = mysql_query("SELECT * FROM liens") or die(mysql_error());
$liste_liens = array();

while ( $donnees = mysql_fetch_assoc($requete))  {
  $liste_liens[$donnees['s']][] = $donnees['e'];
 }

$indices = array_keys($liste_sites);
$op = 0;
$np = 1;

while ($op <> $np)
 {
  $op = $np;
  $np = round(calcul_pagerank( $indices[0]),2);

  foreach ($liste_sites as $site) {
     $objectif = $site['id'];
     $liste_sites[$objectif]['pagerank'] = calcul_pagerank($objectif);
    }
  echo "Tour de mise à jour effectué avec succès.<br /><br />";
 }

foreach ($liste_sites as $site)
 {
  $id = $site['id'];$nouveau_pagerank = $site['pagerank'] ;
  $requete = mysql_query("UPDATE sites SET pagerank=$nouveau_pagerank WHERE id='$id'")
           or die(mysql_error());
 }

echo "Mise à jour du pagerank effectuée avec succès.<br /><br />";
mysql_close($connexion);
  }

?>

<br>Bienvenue sur le programme de calcul DoogleRank.<br>
<form action="dooglerank.php" method="get">
<input value="Démarer le calcul" name="valider" type="submit">
</form>

</body></html>
```


Doogle Rank
Variables globales
global $objectif
id de la page en cours d'analyse, necessaire pour le filtrage
global $liste_sites
Tableau contenant la liste des sites de la forme analogue à la table de la base de données
global $liste_liens
Tableau contenant la liste des liens de la forme analogue à la table de la base de données
Fonctions de filtrage
fonction sous_filtre(variable)
Retourne vrai si variable est égale à la variable globale $objectif
fonction filtre(variable)
Utilise la fonction sous_filtre et renvoie vrai si le tableau envoyé en variable contient au moins une occurence de la variable gloable $objectif.
Fonctions
fonction calcul_pagerank(id)
Calcule le Pagerank du site d'identifiant "id" à partir des informations des tableaux et de la fonction filtre qui détermine la liste des sites ayant des liens vers "id"
fonction get_pagerank(id)
Récupère le Pagerank du site d'identifiant "id" dans le tableau de sites
Programme
Connexion à la base de données
Déclenchement par l'utilisateur
Une requête SQL permet d'initialiser les tableaux :
Pour chaque site
Insertion dans
$liste_sites
Pour chaque liens
Insertion dans
$liste_liens
variable $op = 0
ancien Pagerank du premier site
variable $np = 1
ancien Pagerank du premier site
Tant que
$op
≠
$np
$op
$np
$np
calcul_pagerank (premier site)
Pour chaque site
$liste_sites
est mis à jour par
calcul_pagerank (site)
On met à jour la base de données avec le tableau
$liste_sites
Fermeture de la connexion à la base de données

Nous avons effectué certains tests pour comparer les deux algorithmes.

Calcul du Pagerank : DoogleRank.php		
Ancienne version (requêtes SQL à chaque tour)		**Nouvelle version (calculs dans un tableau)**
Calcul du Pagerank avec deux décimales : (5 tours de calcul) 47 secondes Calcul du Pagerank avec quatre décimales : (2 tours de calcul supplémentaires) 4 secondes supplémentaires **Temps moyen d'un tour de calcul :** **7,286 secondes**		Calcul du Pagerank avec deux décimales : (5 tours de calcul) 36 secondes Calcul du Pagerank avec quatre décimales : (2 tours de calcul supplémentaires) 1 seconde supplémentaires **Temps moyen d'un tour de calcul :** **5,286 secondes**

Les statistiques montrent que l'utilisation d'un tableau fait gagner environ 2 secondes par tour de calculs. Pour seulement 703 pages, 7 tours ont été suffisants, mais à l'échelle du web, de nombreux autres tours seront nécessaires, et cet apport ne sera pas négligeable. Les calculs dans le script évitent les incessantes requêtes à la base de données, et bien que le code source soit moins élégant, il se révèle beaucoup plus efficace.

CONCLUSION

L'avenir des navigateurs

Cependant, le système du Pagerank, aussi bien conçu qu'il soit pour un usage normal, restait perfectible et faillible. Des webmasters peu scrupuleux, afin d'attirer les visiteurs sur leurs sites, gonflaient leur Pagerank à l'aide de milliers de fausses pages contenant des liens vers leur propre site. Google comprit vite qu'il était de la plus haute importance d'empêcher ce genre d'abus. Afin d'améliorer son moteur de recherche, la firme de Mountain View a mis au point des améliorations de son algorithme Pagerank.

Nous avons décidé de ne pas mettre en place ces algorithmes sur notre réplique de Google pour la simple raison qu'ils n'apporteraient pas de résultats significatifs sur une partie si restreinte du web. Cependant, nous avons réfléchi pour chaque cas à des solutions techniques pour implanter ces algorithmes.

L'algorithme HITS (Hyperlinked Induced Topic Search)

Cet algorithme consiste à classer les sites en trois catégories :
- Les sites classiques
- Des sites de **référence** (« authorities ») : ce sont des sites spécialisés dans leurs domaines dont les informations sont considérées comme fiables par vérification humaine.
- Des sites **moyeux** (« hubs ») : véritables annuaires de liens, ils disposent de nombreuses sorties vers les sites de référence, et structurent ainsi la toile du web.

Cette structure permet peut-être de mieux refléter celle d'internet, en privilégiant les sites de référence en résultats de recherche.

Pour l'implémenter, il faudrait ajouter une colonne « Type » à la table SITES, qui contiendrait un entier (0 pour un site classique, 1 pour un authority, 2 pour un hub).
Dans le calcul du Pagerank, on distinguerait les cas, en rendant fixe et supérieur aux autres le Pagerank des authorities, et en donnant plus de poids aux liens sortants d'un hub.
Aucune démonstration mathématique ne peut cependant garantir la convergence d'un tel modèle.

Le Blockrank

Cet algorithme est particulièrement adapté à la structure du web : les pages ne sont pas réparties n'importe comment mais organisées en sites web. Il s'agit alors de calculer le Pagerank en considérant les sites web comme des blocs de pages indissociables et de leur attribuer un Pagerank d'ensemble avant d'affiner le calcul. Les résultats sont très probants.

Dans l'état actuel des choses, notre version de Doogle tourne en réseau local et est donc incapable de repérer les pages provenant du même site (qui ont même nom de domaine). Nous proposons de créer une nouvelle table DOMAINE qui serait reliée à la table SITES par une colonne dans celle-ci. Chaque domaine aurait un Domainerank propre, analogue du Pagerank, et on calculerait le Domainerank dans cette table, à l'aide d'une table DOMAINELIENS. Une nouvelle page de calculs serait nécessaire à la transcription de la table LIENS dans la table DOMAINELIENS, et à la transcription du Domainerank calculé de la table DOMAINE vers la table SITES.

Le pagerank modulaire

Il part du postulat que l'on peut approximer correctement le Pagerank en ne considérant tout d'abord qu'un nombre réduit de sites webs à fort Pagerank, le Hub Set. Les gains ne se montrent pas exceptionnels.

Pour ce faire, il faudrait probablement créer une colonne « Hub Set » sur la table SITES, ainsi qu'une nouvelle page d'initialisation qui calcule les Pagerank selon la méthode la plus longue, et selectionne ensuite les Pagerank les plus élevés pour les inscrire dans le Hub Set (un boléen dans la base de données).

Le pagerank thématique (Topic Sensitive Pagerank)

Il s'agit d'influencer le calcul du Pagerank en fonction de la recherche de l'utilisateur. Il faut pour cela définir arbitrairement plusieurs thèmes, et repérer les principaux sites de référence en cette matière (par une analyse de fréquence d'apparition d'un mot par exemple). Ces sites là seront crédités d'un Pagerank plus élevé pour le calcul qui s'ensuit.

Nous suggérons l'ajout d'une table THEMES, avec un mot clé pour chaque thème et une liste de sites web qui sont désignées comme référence (éventuellement mise à jour par un scan des fréquences des mots clés). Il s'agirait ensuite de rattacher la requête de l'utilisateur à l'un des thèmes (en regardant par exemple dans quel thème sont les pages où ce mot apparaît le plus fréquemment), et de recalculer par la suite le Pagerank en prenant garde à laisser un Pagerank supérieur aux sites concernés. Nous pouvons même envisager l'implantation d'une telle structure non plus pour les thèmes, mais simplement pour les mots, ce qui dispenserait de l'utilisation d'une table supplémentaire.

Bibliographie

Images :

Illustration du Pagerank en en-tête

Felipe Micaroni Lalli pour wikipedia

Photographies de Mr Brin et Mr Page, et éléments biographiques

http://fr.wikipedia.org/wiki/Sergey_Brin

http://fr.wikipedia.org/wiki/Larry_Page

Logo MySQL :

http://www.cnp-x.com/mysql/graphics/MySQL_logo.gif

Logo PHP :

http://www.elroubio.net/logo_elePHPant_special_presse.htm

Informations sur internet ou les moteurs de recherche :

Population d'internet

Jesse Alpert & Nissan Hajaj, official Google Blog, 25/7/08

Principales sources sur le fonctionnement de Google

http://www.optimisation.ca/engins-de-recherche.html

http://www.webmaster-hub.com/publication/Quelques-pistes-pour-comprendre-le.html

Le futur des moteurs de recherches

http://www.webmaster-hub.com/publication/Vers-un-moteur-de-recherche,54.html?var_recherche=kaltix

http://www.webmaster-hub.com/publication/Quelques-pistes-pour-comprendre-le,100.html

Productions :

Recherches non organisées

Google, septembre 2008

Pages internet aspirées à partir du site web choisi aléatoirement :

Site personnel de Joël Le Roux, Professeur à l'Ecole Polytechnique de l'Université de Nice, France

http://www.polytech.unice.fr/~leroux

Etude théorique du Pagerank (principales sources) :

Quadrature - avril/juin 2008

A note on the PageRank algorithm

Huan Sun, Yimin Wei

School of Mathematical Sciences, Fudan University and Key Laboratory of Mathematics for Nonlinear Sciences,

Ministry of Education, Shanghai 200433, PR China

THE EFFECT OF NEW LINKS ON GOOGLE PAGERANK

Konstantin Avrachenkov *INRIA Sophia Antipolis, France*

Nelly Litvak *Department of Applied Mathematics, University of Twente, Enschede, The Netherlands*

The Condition Number of the PageRank Problem

Sepandar D. Kamvar and Taher H. Haveliwala

Stanford University

***Adaptive methods for the computation* of PageRank**

Sepandar Kamvar a, Taher Haveliwala b, Gene Golub a

a- *Scientific Computing and Computational Mathematics*

b- *Department of Computer Science*

Stanford University, Stanford, CA 94305, USA

***An Analytical Comparison of Approaches to* Personalizing PageRank**

Taher Haveliwala, Sepandar Kamvar and Glen Jeh

Stanford University

Aide externe :

Conseils et avis sur l'algorithme de recherche :

Mr Dominique Mallet

Professeur de Mathématiques et d'informatique au lycée Henri Wallon, Valenciennes

Convergence mathématique du Pagerank :

Mr Jean Zurek
Professeur de Mathématiques au lycée Henri Wallon, Valenciennes

Éclairage professionnel et technique :

Mr Jean Deruelle
Senior Software Engineer chez **JBoss**, a division of **Red Hat**.
Project Lead of the **Mobicents Sip Servlets** Project

Logiciels utilisés :

Rédaction : Open Office
Editeur PHP : Notepad ++
Retouche et créations graphiques, schémas : Paint, Inkscape, Gimp