Remarques sur TSP ; PageRank ; résumé du cours

CM nº12 — Algorithmique (AL5)

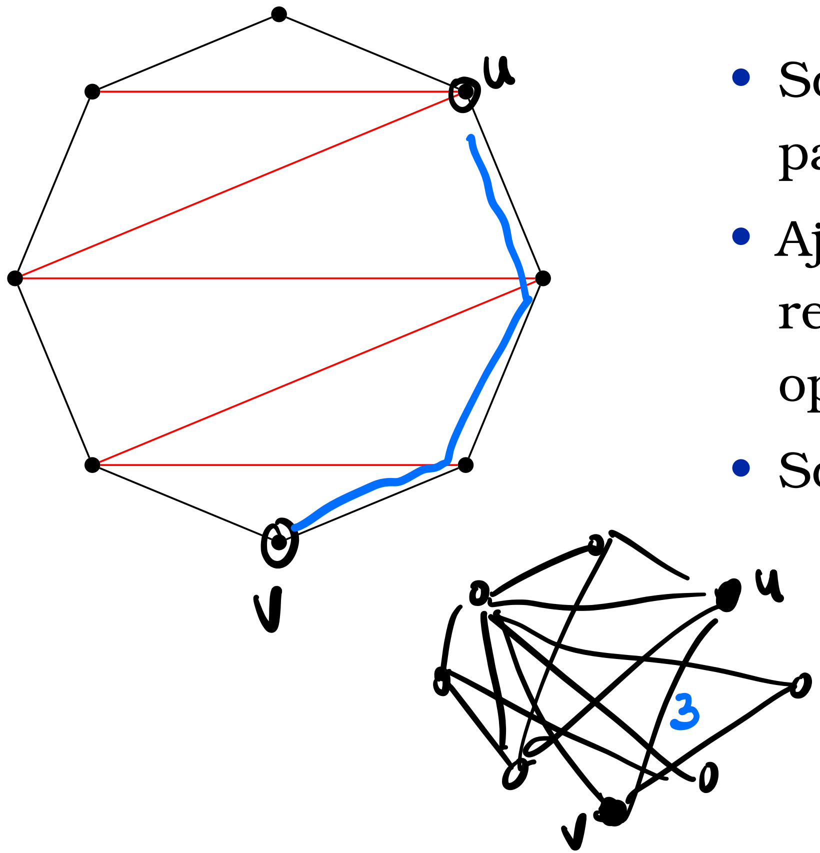
Matěj Stehlík

15/12/2023

Rappel : le facteur d'approximation de Christofides

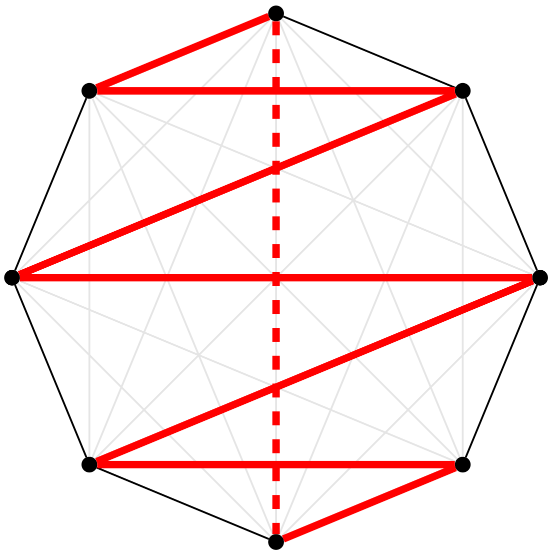
- Nous avons vu que l'algorithme de Christofides est un algorithme de 1.5-approximation pour le problème du voyageur de commerce.
- Mais le facteur d'approximation pourrait-il être meilleur que 1.5 ?
- Malheureusement, il n'est pas difficile de trouver des exemples où Christofides trouve une solution de poids $1.5 \cdot \mathrm{OPT}$.
- (De même, on peut trouver des exemples où Double-Tree trouve une solution de poids $2 \cdot \mathrm{OPT}$.)

Exemple où Christofides trouve une solution de poids $1.5 \cdot \mathrm{OPT}$



- Soit C_n un cycle à n sommets, avec n pair.
- Ajoutons un chemin P en zig-zag qui relie deux sommets diamétralement opposés de C_n.
- Soit H le graphe résultant.

Exemple où Christofides trouve une solution de poids $1.5 \cdot \mathbf{OPT}$



- Soit G le graphe complet à n-sommets avec $V(G) = V(H)$, avec pondération $w_{uv} = \mathrm{dist}_H(u, v)$.
- On a $\mathrm{OPT} = n$.
- Supposons que T est la chaîne P.
- Alors, J est une arête entre deux sommets diamétralement opposés, de poids $n/2$.
- La solution trouvée par l'algorithme de Christofides est alors de poids $n - 1 + n/2 = 3n/2 - 1$.
- Cela tend vers $3n/2$ lorsque $n \to \infty$.

Pourquoi classer les pages web par ordre d'importance ?

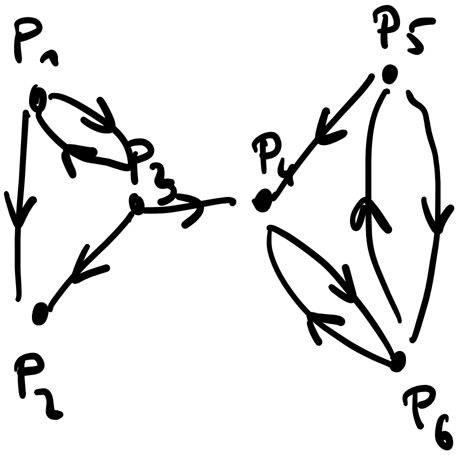
- Que se passe-t-il lorsque vous soumettez une requête à Google ?
- Une liste de pages web pertinentes est produite, sur la base des mots de votre requête.
- Cette liste de pages web est classée de manière à ce que les pages les plus utiles pour vous figurent en tête de liste.
- Le succès de Google dans cette deuxième étape a largement contribué à en faire le moteur de recherche le plus populaire au monde.
- Le score de popularité d'une page web est appelé le PageRank.

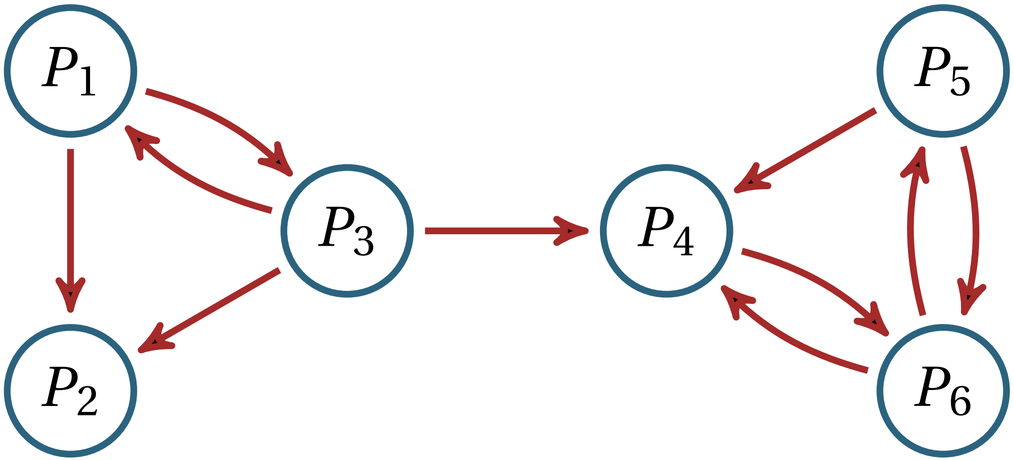
L’intuition pour le PageRank

- Un internaute imaginaire navigue au hasard sur le web.
- L’internaute commence par une page web aléatoire.
- Chaque fois qu’il visite une page web, il choisit au hasard un hyperlien sur cette page et le suit.
- L’internaute continue ainsi indéfiniment.
- À la limite, la proportion $r(P_i)$ du temps que l’internaute passe sur une page particulière P_i est une mesure de l’importance de la page P_i.

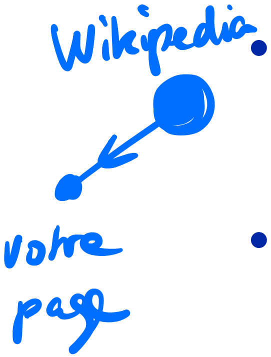
Le graphe du Web

- On peut représenter le Web par un graphe orienté G :
  - les sommets de G sont les pages web $P_1, P_2, \ldots$
  - (P_i, P_j) est un arc de G ssi il y a un hyperlien de la page P_i vers la page P_j

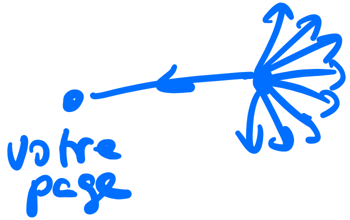




- Le but est d’attacher à chaque page web un nombre qui représente son importance.
- Comment mesurer l’importance d’une page ?
- Un lien de votre page vers ma page est une approbation de ma page.
- Plus ma page a de liens, plus elle est importante.
- Toutefois, certains liens ont plus de valeur que d’autres.



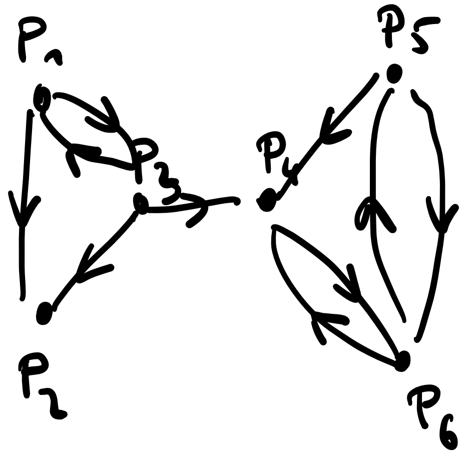
  - Un lien vers ma page provenant d’une page très importante a plus de valeur. Plus une page est importante, plus elle doit contribuer à l’importance des pages vers lesquelles elle renvoie.
  - Un lien vers ma page à partir d’une page comportant un grand nombre de liens sortants n’a pas autant de valeur. Plus une page a de liens externes, moins la recommandation fournie par chacun de ces liens a de valeur.



- Le PageRank de la page P_i est noté $r(P_i)$. Il s'agit d'un nombre qui mesure l'importance de la page P_i.
- Le nombre de liens sortants de la page P_i est noté $d^+(P_i)$.
- $d^+(P_i)$ est égal à la somme des entrées de la $i$ème ligne de la matrice d'adjacence A.
- Chaque page P_j qui renvoie à la page P_i apporte une partie de son importance $r(P_j)$ à l'importance de la page P_i.
- Plus une telle page a de liens sortants, moins elle devrait contribuer à l'importance de la page P_i.
- Par conséquent, si la page P_i a un lien entrant depuis la page P_j, nous dirons que P_j contribue $\frac{r(P_j)}{d^+(P_j)}$ au PageRank $r(P_i)$ de la page P_i.
- En additionnant toutes les contributions des pages ayant un lien vers P_i, on obtient le PageRank de P_i.

- Pour le "mini web" de six pages, cela conduit aux équations de PageRank suivantes :

$$r(P_1) = \frac{r(P_3)}{3}$$
$$r(P_2) = \frac{r(P_1)}{2} + \frac{r(P_3)}{3}$$
$$r(P_3) = \frac{r(P_1)}{2}$$
$$r(P_4) = \frac{r(P_3)}{3} + \frac{r(P_5)}{2} + \frac{r(P_6)}{2}$$
$$r(P_5) = \frac{r(P_6)}{2}$$
$$r(P_6) = r(P_4) + \frac{r(P_5)}{2}$$



- Nous recherchons un vecteur PageRank

$$v = (r(P_1), r(P_2), r(P_3), r(P_4), r(P_5), r(P_6))$$

qui satisfait à ces six équations du PageRank.

- En outre, comme $r(P_i)$ doit donner la proportion de temps que l'internaute passe sur la page P_i, le vecteur v doit être un vecteur de probabilité.
- C'est-à-dire, il faut que $r(P_i) \geq 0$ pour chaque page P_i, et

$$\sum_{i=1}^{6} r(P_i) = 1.$$

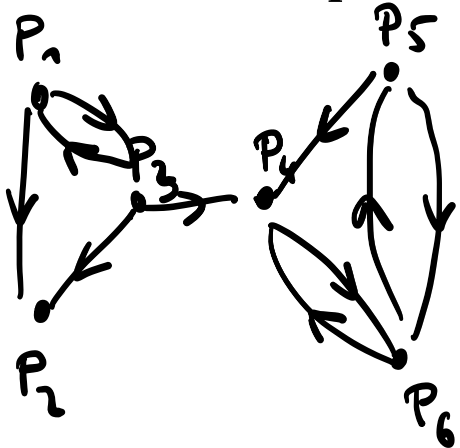
- Les six équations de PageRank présentées précédement peuvent être résumées en une seule équation matricielle.
- La matrice d'adjacance A peut être normalisée en divisant chaque entrée de la matrice A par la somme de sa ligne.
- La somme de chaque ligne non nulle est maintenant égale à 1.
- La matrice résultante B est appelée matrice des hyperliens normalisée (par ligne) du graphe G.

- Si on note l'ensemble des pages ayant un hyperlien vers P_i par $N^-(P_i)$, alors chaque équation de PageRank peut s'écrire en notation de sommation comme suit

$$r(P_i) = \sum_{P_j \in N^-(P_i)} \frac{r(P_j)}{d^+(P_j)}.$$

$$A = \begin{array}{cc} & \begin{array}{cccccc} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \end{array} \\ \begin{array}{c} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{array} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array} \qquad B = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

- Les six équations du PageRank sont équivalentes à l'équation matricielle $v = vB$



- Il serait facile de résoudre les six équations de PageRank à la main.
- Il existe également une méthode systématique pour résoudre un tel système d'équations simultanées (l'algorithme de Gauss) qui peut être exécutée par un ordinateur.
- Cependant, la détermination des PageRank pour l'ensemble du World Wide Web implique un système de plusieurs milliards d'équations de PageRank, et il n'est pas possible d'essayer de résoudre ces équations de PageRank directement, même à l'aide d'un ordinateur.

- L'équation matricielle $v = vB$ peut être résolue de manière itérative, c'est-à-dire en calculant une séquence d'approximations du vecteur solution v.
- Pour l'approximation initiale $v^{(0)}$, ils considèrent que toutes les pages ont le même PageRank. Pour notre exemple du "mini web", cela donne

$$v^{(0)} = \left(\tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}\right)$$

- Il s'agit de la 0e approximation du vecteur PageRank v.
- Les approximations successives sont obtenues en multipliant à droite par la matrice B :

$$v^{(1)} = v^{(0)}B$$
$$v^{(2)} = v^{(1)}B \ldots$$

- On espère que la séquence $v^{(0)}, v^{(1)}, \ldots$ converge vers un vecteur de probabilité v avec $v = vB$.
- Deux questions importantes se posent :
  1. Existe-t-il un théorème garantissant que la séquence $v^{(0)}, v^{(1)}, \ldots$ converge vers un vecteur v ?
  2. Si la séquence $v^{(0)}, v^{(1)}, \ldots$ converge vers un vecteur v, ce dernier sera-t-il nécessairement un vecteur de probabilité ?

- Nous traiterons d'abord la deuxième question.
- Dans le cas de notre "mini web" de six pages, nous avons $v^{(0)} = \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)$.
- En utilisant une tableur, nous pouvons facilement appliquer la formule itérative $v^{(k+1)} = v^{(k)}B$ pour calculer $v^{(1)}, v^{(2)}, \ldots$

	P_1	P_2	P_3	P_4	P_5	P_6
v^(0)	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667
v^(1)	0.05555556	0.13888889	0.08333333	0.22222222	0.08333333	0.25000000
v^(2)	0.02777778	0.05555556	0.02777778	0.19444444	0.12500000	0.26388889
v^(3)	0.00925926	0.02314815	0.01388889	0.20370370	0.13194444	0.25694444
v^(4)	0.00462963	0.00925926	0.00462963	0.19907407	0.12847222	0.26967593
v^(5)	0.00154321	0.00385802	0.00231481	0.20061728	0.13483796	0.26331019
v^(6)	0.00077160	0.00154321	0.00077160	0.19984568	0.13165509	0.26803627
v^(7)	0.00025720	0.00064300	0.00038580	0.20010288	0.13401813	0.26567323
v^(8)	0.00012860	0.00025720	0.00012860	0.19997428	0.13283661	0.26711195
v^(9)	0.00004287	0.00010717	0.00006430	0.20001715	0.13355597	0.26639259
v^(10)	0.00002143	0.00004287	0.00002143	0.19999571	0.13319629	0.26679513
v^(11)	0.00000714	0.00001786	0.00001072	0.20000286	0.13339757	0.26659386
v^(12)	0.00000357	0.00000714	0.00000357	0.19999929	0.13329693	0.26670164
v^(13)	0.00000119	0.00000298	0.00000179	0.20000048	0.13335082	0.26664775
v^(14)	0.00000060	0.00000119	0.00000060	0.19999988	0.13332388	0.26667589
v^(15)	0.00000020	0.00000050	0.00000030	0.20000008	0.13333794	0.26666182
v^(16)	0.00000010	0.00000020	0.00000010	0.19999998	0.13333091	0.26666905
v^(17)	0.00000003	0.00000008	0.00000005	0.20000001	0.13333453	0.26666543
v^(18)	0.00000002	0.00000003	0.00000002	0.20000000	0.13333272	0.26666728
v^(19)	0.00000001	0.00000001	0.00000001	0.20000000	0.13333364	0.26666636
v^(20)	0.00000000	0.00000001	0.00000000	0.20000000	0.13333318	0.26666682
v^(21)	0.00000000	0.00000000	0.00000000	0.20000000	0.13333341	0.26666659
v^(22)	0.00000000	0.00000000	0.00000000	0.20000000	0.13333329	0.26666671
v^(23)	0.00000000	0.00000000	0.00000000	0.20000000	0.13333335	0.26666665
v^(24)	0.00000000	0.00000000	0.00000000	0.20000000	0.13333332	0.26666668
v^(25)	0.00000000	0.00000000	0.00000000	0.20000000	0.13333334	0.26666666
v	0	0	0	1/5	2/15	4/15

- Nous constatons que le vecteur limite v semble être $v = \left(0, 0, 0, \frac{1}{5}, \frac{2}{15}, \frac{4}{15}\right)$
- En effet, il est facile de vérifier que ce choix de v satisfait l'équation matricielle $v = vB$.
- Malheureusement, comme la somme des coordonnées est $\frac{3}{5}$ au lieu de 1, il ne s'agit pas d'un vecteur de probabilité !
- Le problème est causé par la rangée de zéros dans la matrice B.
- Cette ligne de zéros correspond au fait que P_2 est un puits, c'est-à-dire qu'il n'a pas d'arcs sortants.
- Les puits sont très courants sur le Web (par exemple : fichiers images, documents PDF, etc.) et posent un problème à notre internaute aléatoire.
- Lorsqu'il entre un puits, il ne peut jamais sortir !

- Pour surmonter ce problème, lorsque l'internaute entre dans un puits, il peut sauter à n'importe quelle page au hasard.
- Cela correspond au remplacement de chaque ligne de 0 dans la matrice B par une ligne de $\frac{1}{n}$, où n est le nombre total de sommets dans le graphe.
- Cette nouvelle matrice S est appelée la matrice stochastique du graphe G, car chaque ligne somme à 1.
- Pour notre exemple, cette transformation est illustrée ci-dessous :

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \qquad S = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

- En utilisant une tableur, nous pouvons facilement appliquer la formule itérative $v^{(k+1)} = v^{(k)}S$ pour calculer $v^{(1)}, v^{(2)}, \ldots$

	P_1	P_2	P_3	P_4	P_5	P_6
v^(0)	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667
v^(1)	0.08333333	0.16666667	0.11111111	0.25000000	0.11111111	0.27777778
v^(2)	0.06481481	0.10648148	0.06944444	0.25925926	0.16666667	0.33333333
v^(3)	0.04089506	0.07330247	0.05015432	0.29089506	0.18441358	0.36033951
v^(4)	0.02893519	0.04938272	0.03266461	0.30131173	0.19238683	0.39531893
v^(5)	0.01911866	0.03358625	0.02269805	0.31297154	0.20588992	0.40573560
v^(6)	0.01316372	0.02272305	0.01515704	0.31897648	0.20846551	0.42151420
v^(7)	0.00883952	0.01542138	0.01036904	0.32382938	0.21454428	0.42699641
v^(8)	0.00602658	0.01044634	0.00698999	0.32679692	0.21606843	0.43367174
v^(9)	0.00407105	0.00708434	0.00475434	0.32894114	0.21857693	0.43657219
v^(10)	0.00276550	0.00480103	0.00321625	0.33034006	0.21946682	0.43941033
v^(11)	0.00187226	0.00325501	0.00218292	0.33131083	0.22050534	0.44087365
v^(12)	0.00127014	0.00220627	0.00147863	0.33195963	0.22097932	0.44210600
v^(13)	0.00086059	0.00149566	0.00100278	0.33240325	0.22142071	0.44281701
v^(14)	0.00058354	0.00101383	0.00067957	0.33270240	0.22165778	0.44336288
v^(15)	0.00039550	0.00068726	0.00046074	0.33290583	0.22185041	0.44370026
v^(16)	0.00026812	0.00046587	0.00031229	0.33304346	0.22196467	0.44394558
v^(17)	0.00018174	0.00031580	0.00021171	0.33313687	0.22205043	0.44410344
v^(18)	0.00012320	0.00021407	0.00014351	0.33320014	0.22210436	0.44421472
v^(19)	0.00008351	0.00014512	0.00009728	0.33324305	0.22214304	0.44428800
v^(20)	0.00005661	0.00009837	0.00006594	0.33327213	0.22216819	0.44433876
v^(21)	0.00003838	0.00006668	0.00004470	0.33329185	0.22218577	0.44437262
v^(22)	0.00002601	0.00004520	0.00003030	0.33330521	0.22219742	0.44439585
v^(23)	0.00001763	0.00003064	0.00002054	0.33331427	0.22220546	0.44441146
v^(24)	0.00001195	0.00002077	0.00001392	0.33332041	0.22221083	0.44442211
v^(25)	0.00000810	0.00001408	0.00000944	0.33332457	0.22221451	0.44442929
v	0	0	0	1/3	2/9	4/9

4 4 4 2 3 1

- Le vecteur limite v semble être $v = \left(0, 0, 0, \frac{1}{3}, \frac{2}{9}, \frac{4}{9}\right)$
- Il s'agit bien d'un vecteur de probabilité, car $\frac{1}{3} + \frac{2}{9} + \frac{4}{9} = 1\,!$
- Le remplacement des lignes de zéros a donc résolu le problème causé par les sommets puits
- Si on utilise la matrice stochastique S au lieu de la matrice B, le vecteur limite v, lorsqu'il existe, sera un vecteur de probabilité.
- Cependant, même lorsque le vecteur limite v existe, notre exemple illustre un autre problème.
- Nous souhaitons utiliser les PageRanks de nos six pages web pour produire un classement des pages.

- Malheureusement, les PageRanks calculés ci-dessus ne nous permettent pas de distinguer les pages P_1, P_2 et P_3, car elles ont toutes un PageRank égal à 0.
- Le mieux que nous puissions faire est de classer les pages $(4, 4, 4, 2, 3, 1)$.
- Afin d'éliminer le problème des pages multiples ayant un PageRank de 0, nous ajoutons une troisième question à notre liste :
  3. Si la séquence $v^{(0)}, v^{(1)}, v^{(2)}, v^{(3)}, \ldots$ converge vers un vecteur v, comment pouvons-nous garantir que v est un vecteur positif, c'est-à-dire que chaque entrée de v est positive ?

- Il existe une modification simple de la matrice S qui répondra simultanément aux questions 1 et 3 : nous aurons la garantie que la séquence des vecteurs $v^{(0)}, v^{(1)}, v^{(2)}, v^{(3)}, \ldots$ converge vers un vecteur positif v.
- Cette modification peut être justifiée par le comportement de notre internaute aléatoire.
- L'internaute se lasse de suivre des hyperliens et demande une page web complètement aléatoire.
- Une fois sur cette page, il continuera à suivre les hyperliens jusqu'à ce qu'il s'ennuie à nouveau.

- Si la proportion du temps pendant lequel l'internaute continue à suivre les hyperliens est d, alors la proportion du temps pendant lequel il passe à une autre page choisie au hasard est $1-d$.
- Le nombre d, appelé *facteur d'amortissement*, est choisi de manière à ce que $0 < d < 1$.
- Pour représenter ce nouveau comportement de ~~surf~~ navigation pour notre exemple de six pages G, nous devons remplacer chaque ligne r de la matrice S par

$$dr + (1-d)\left(\tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}, \tfrac{1}{6}\right).$$

- Cela donne une nouvelle matrice $M = dS + \left(\frac{1-d}{n}\right) J$, où J est la matrice $n \times n$ avec toutes les entrées égales à 1.

- Dans leur article original de 1998, Brin et Page indiquent qu'ils fixent habituellement $d = 0.85$.
- Avec ce choix, la matrice devient

$$M = \begin{pmatrix} \frac{1}{40} & \frac{9}{20} & \frac{9}{20} & \frac{1}{40} & \frac{1}{40} & \frac{1}{40} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{37}{120} & \frac{37}{120} & \frac{1}{40} & \frac{37}{120} & \frac{1}{40} & \frac{1}{40} \\ \frac{1}{40} & \frac{1}{40} & \frac{1}{40} & \frac{1}{40} & \frac{1}{40} & \frac{9}{20} \\ \frac{1}{40} & \frac{1}{40} & \frac{1}{40} & \frac{9}{20} & \frac{1}{40} & \frac{9}{20} \\ \frac{1}{40} & \frac{1}{40} & \frac{1}{40} & \frac{9}{20} & \frac{9}{20} & \frac{1}{40} \end{pmatrix}$$

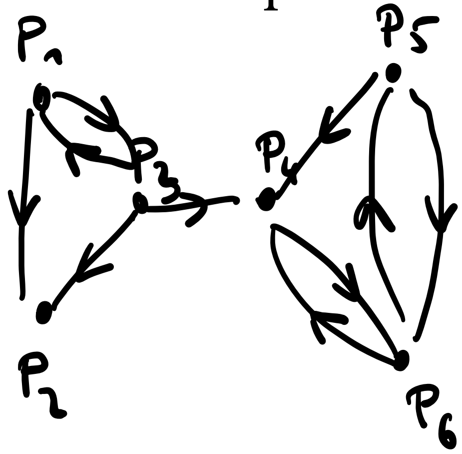
- Nous pouvons à nouveau répéter l'itération, en utilisant cette fois la matrice M (avec le facteur d'amortissement $d = 0.85$).
- C'est-à-dire, nous utilisons maintenant la formule $v^{(k+1)} = v^{(k)}M$ pour calculer $v^{(1)}, v^{(2)}, \ldots$
- En utilisant un tableur, nous obtenons le résultat suivant

	P_1	P_2	P_3	P_4	P_5	P_6
v^(0)	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667	0.16666667
v^(1)	0.09583333	0.16666667	0.11944444	0.23750000	0.11944444	0.26111111
v^(2)	0.08245370	0.12318287	0.08934028	0.24418981	0.15958333	0.30125000
v^(3)	0.06776399	0.10280681	0.07749373	0.26361815	0.17048216	0.31783517
v^(4)	0.06152086	0.09032055	0.06836399	0.26905572	0.17464424	0.33609464
v^(5)	0.05716521	0.08331157	0.06394177	0.27422924	0.18063563	0.34071657
v^(6)	0.05491931	0.07921452	0.06109769	0.27649400	0.18160702	0.34666747
v^(7)	0.05353307	0.07687377	0.05956276	0.27804972	0.18355573	0.34842494
v^(8)	0.05276657	0.07551812	0.05864201	0.27885835	0.18397105	0.35024390
v^(9)	0.05231364	0.07473943	0.05812419	0.27935499	0.18455206	0.35091570
v^(10)	0.05205661	0.07428990	0.05782138	0.27963040	0.18472726	0.35147445
v^(11)	0.05190713	0.07403118	0.05764846	0.27979285	0.18490105	0.35171933
v^(12)	0.05182148	0.07388201	0.05754828	0.27988514	0.18496847	0.35189462
v^(13)	0.05177196	0.07379609	0.05749075	0.27993878	0.18502183	0.35198059
v^(14)	0.05174349	0.07374658	0.05745753	0.27996952	0.18504620	0.35203668
v^(15)	0.05172707	0.07371805	0.05743842	0.27998729	0.18506302	0.35206616
v^(16)	0.05171761	0.07370161	0.05742739	0.27999751	0.18507151	0.35208437
v^(17)	0.05171216	0.07369214	0.05742105	0.28000340	0.18507692	0.35209433
v^(18)	0.05170902	0.07368668	0.05741739	0.28000680	0.18507981	0.35210030
v^(19)	0.05170721	0.07368354	0.05741528	0.28000876	0.18508158	0.35210365
v^(20)	0.05170616	0.07368173	0.05741406	0.28000988	0.18508255	0.35210561
v^(21)	0.05170556	0.07368068	0.05741336	0.28001053	0.18508313	0.35210673
v^(22)	0.05170522	0.07368008	0.05741296	0.28001091	0.18508346	0.35210738
v^(23)	0.05170502	0.07367973	0.05741273	0.28001112	0.18508365	0.35210775
v^(24)	0.05170490	0.07367953	0.05741259	0.28001125	0.18508376	0.35210797
v^(25)	0.05170484	0.07367942	0.05741252	0.28001132	0.18508382	0.35210809
v	0.0517	0.0737	0.0574	0.2800	0.1851	0.3521

- En arrondissant à la quatrième décimale, le vecteur de PageRank pour notre exemple est

$$v = (0.0517, 0.0737, 0.0574, 0.2800, 0.1851, 0.3521).$$

- Cela nous permet de classer nos six pages web dans l'ordre du plus important au moins important : P_6, P_4, P_5, P_2, P_3, P_1.



- Le calcul des PageRanks pour notre exemple implique une matrice 6×6.
- Cependant, le calcul par Google des PageRanks pour l'ensemble du World Wide Web implique une matrice de plusieurs milliards sur plusieurs milliards !
- La raison pour laquelle ce calcul est possible est que la matrice M est basée sur la matrice d'adjacance A, qui est peu dense (la plupart de ses entrées sont des zéros).
- S'il y a n pages web, la matrice A a n^2 entrées.
- En moyenne, une page web n'a que 10 liens sortants, et il y a donc environ $10n$ entrées non nulles dans la matrice A.
- Comme le nombre de pages web n est extrêmement grand, le nombre $10n$ est très inférieur à n^2.

- Le calcul des PageRanks à partir de la matrice M repose sur plusieurs théorèmes mathématiques fondamentaux :
- Puisque M est une matrice stochastique positive (c'est-à-dire que chaque entrée est positive et que chaque ligne est égale à 1), le théorème de Perron–Frobenius garantit qu'il existe un vecteur unique de probabilité positive v qui satisfait l'équation $v = vM$.
- La théorie des chaînes de Markov garantit que :
  - que la séquence $v^{(0)}, v^{(1)}, v^{(2)}, v^{(3)}, \ldots$ converge vers v, quel que soit le choix du vecteur de départ $v^{(0)}$;
  - pour produire des scores PageRank avec une précision d'environ m chiffres, le nombre d'itérations doit être de l'ordre de

$$\frac{m}{\log_{10}(\frac{1}{d})}.$$

- Apparemment, Google utilise entre 50 et 100 itérations, ce qui garantit une précision de 3 à 7 chiffres.

Résumé du cours (1/4)

CM1 Graphes et complexité d'algorithmes

- Représentation matricielle et par listes
- Lemme des poignées de main
- Notation grand O

CM2 Connexité, arbres, parcours en largeur

- Caractérisation des arbres
- Parcours en largeur (BFS)
- Implémentation du BFS avec files

CM3 Parcours en profondeur

- Version récursive
- Version avec piles
- Prévisite et postvisite
- Classification des arcs (arbre/avant/retour/transverse)
- DFS trouve un arc retour ssi il existe un circuit
- Graphes orientés acycliques (DAG)

Résumé du cours (2/4)

CM4 Tri topologique et composantes fortement connexes

- Tri topologique : algorithme de Kahn ou DFS avec postvisite
- Calcul des composantes fortement connexes

CM5 Plus court chemin

- Algorithme de Dijkstra
- Implémentation de Dijkstra avec files de priorité
- Algorithme de Bellman–Ford
- Détection de cycles négatifs

CM6 Plus court chemin entre toutes les paires de sommets

- Plus court chemin dans les DAG (en faisant un tri topologique)
- Algorithme de Floyd–Warshall
- Algorithme de Johnson

CM7 Arbres couvrants minimaux

- Algorithme de Kruskal
- Implémentation de Kruskal avec union-find
- Algorithme de Prim
- Implémentation de Prim avec files de priorité

Résumé du cours (3/4)

CM8 Flot-max / coupe-min

- Dualité faible, certificat d'optimalité
- Graphe résiduel
- Algorithme de Ford–Fulkerson
- Théorème de flot max / coupe min
- Trouver une coupe min à partir d'un flot max

CM9 Couplages

- Théorème de Berge (couplage max ssi pas de chaîne augmentante)
- Application de Ford–Fulkerson aux couplages
- Théorème de Kőnig ($\tau(G) = \nu(G)$ si G est biparti)
- Lemme des mariages
- Théorème de Tutte
- Mariages stables
- Algorithme de Gale-Shapley

Résumé du cours (4/4)

CM10 Problème du postier et du voyageur de commerce ; algorithmes d'approximation

- Théorème d'Euler
- Algorithme de Hierholzer
- Chaînes eulériennes
- Problème du postier chinois
- Algorithme pour résoudre le problème du postier chinois
- Algorithmes d'approximation
- Algorithme de 2-approximation pour VERTEX COVER
- Problème du voyageur de commerce
- Algorithme Double-Tree
- Algorithme de Christofides

Liste des algorithmes étudiés (1/2)

Algorithme	Objectif	Complexité
BFS	Parcours de graphe	$O(m+n)$
DFS	Parcours de graphe	$O(m+n)$
Kahn	Tri topologique	$O(m+n)$
Kosaraju	Comp. fortement connexes	$O(m+n)$
Dijkstra	Plus court chemin (poids ≥ 0)	$O(m+n\log n)$
Bellman–Ford	Plus court chemin	$O(mn)$
Floyd–Warshall	PCC entre toutes les paires	$O(n^3)$
Johnson	PCC entre toutes les paires	$O(mn+n^2\log n)$
Kruskal	Arbre couvrant minimal	$O(m\log n)$
Prim	Arbre couvrant minimal	$O(m+n\log n)$

Liste des algorithmes étudiés (2/2)

Algorithme	Objectif	Complexité
Ford–Fulkerson	Flot maximum	$O(mC)$
Edmonds–Karp	Flot maximum	$O(m^2n)$
Edmonds [1]	Couplage de poids maximum	$O(mn^2)$
Gale–Shapley	Mariages stables	$O(n^2)$
Hierholzer	Cycle eulérien	$O(m)$
Edmonds–Johnson	Tour de postier chinois	$O(n^3)$
Christofides	Cycle hamiltonien de poids au plus $1.5 \cdot \mathrm{OPT}$	$O(n^3)$

1. Pas présenté ; utilisé comme boîte noire