CS9668/4438 Internet Algorithmics
Assignment 5
For Undergraduate Students
Due December 7

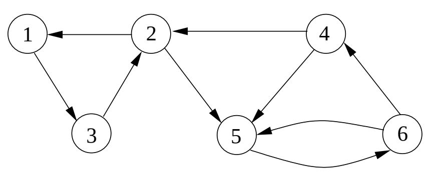
1. • *(15 marks) Explain how to compute the address that processor p must store in the i-th entry of its finger table, given that it knows the address of a processor q.*

   The address that must be stored in the i-th entry of the finger table of processor p is the successor of $h_p(p) + 2^i$. Hence, processor p sends to processor q a <LOOKUP,$h_p(p) + 2^i$> message requesting to find the successor of $h_p(p) + 2^i$. Note that the algorithm that processor q and all other processors in the system use to find the successor of $h_p(p) + 2^i$ is the same algorithm in assignment 4 that is used to look for a document with key $h_p(p) + 2^i$, except that instead of returning to p the document with this key the algorithm returns the address of the processor that must store the key.
   Therefore, each processor p_i that receives a <LOOKUP,value> message scans its finger table to find the closest processor p' that clockwise precedes `value` in the ring of identifiers. If p' is the successor of p_i, then p' is the successor of `value` and thus a message is sent back to p containing the address of p'. If p' is not the successor of p' then p_i sends a <LOOKUP,value> message to p'.

   • *(30 marks) Write a program in Java using the simulator for distributed systems that computes the addresses that must be stored in the finger table of a processor p that has just joined the system.*

   The algorithm can be downloaded from the course's website.

2. *(8 marks) Compute the (simple) page rank of every node in the following graph. Indicate which method you used to compute the page rank.*

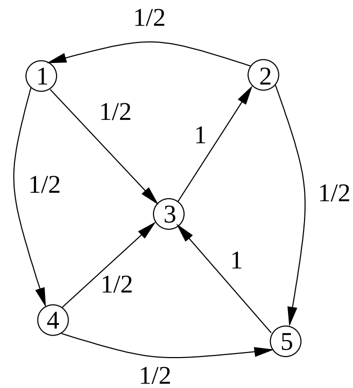


The equations defining the simple page rank for this graph are the following:

$$\begin{aligned}
r(1) &= \frac{1}{2}r(2) && (1)\\
r(2) &= r(3) + \frac{1}{2}r(4) && (2)\\
r(3) &= r(1) && (3)\\
r(4) &= \frac{1}{2}r(6) && (4)\\
r(5) &= \frac{1}{2}r(2) + \frac{1}{2}r(4) + \frac{1}{2}r(6) && (5)\\
r(6) &= r(5) && (6)\\
& && (7)
\end{aligned}$$

Using Gaussian elimination we get the following solution: $r(1) = r(3) = \frac{1}{14}$, $r(2) = r(4) = \frac{1}{7}$, and $r(5) = r(6) = \frac{2}{7}$.

3. *(7 marks) Compute the stationary distribution of the following Markov chain. Explain how you computed the stationary distribution.*



Using the power iteration algorithm we get the following values for the stationary distribution after 69 iterations (a Java implementation of the power iteration algorithm for answering this question can be downloaded from the course's website):

$\pi(0) = 0.148, \pi(1) = 0.296, \pi(2) = 0.296, \pi(3) = 0.074, \pi(4) = 0.185.$

An exact solution is obtained by writing the equations for the stationary distribution and solving the corresponding system of linear equations:

$\pi(0) + \frac{4}{27}, \pi(1) = \frac{8}{27}, \pi(2) = \frac{8}{27}, \pi(3) = \frac{2}{27}, \pi(4) = \frac{5}{27}.$

4. *In class we talked about the problem of ranking pages to be returned by web search engines. Ideally high-quality content pages should be assigned high ranks, and low-quality content pages should be assigned low ranks.*

   Let $r(i)$ denote the page rank of page i, and let n be the total number of pages in the Web graph. Consider the following 3 ways of defining $r(i)$:

   a. *(5 marks) $r(i) = 1/n$.*

      With this definition the page rank exists, but each page gets assigned the same rank, so it does not differentiate between low quality and high quality content.

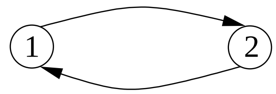
   b. *(10 marks) $r(i) = \text{maximum}\{r(j) \mid j \in B(i)\}$, where $B(i)$ is the set of pages that have hyperlinks pointing to page i.*

      Consider any pair of pages p_i, p_j. Since the graph is strongly connected, there is a path from p_i to p_j and a path from p_j to p_i. These two paths form a cycle: $p_i, p_{i,1}, p_{i,2}, \ldots, p_j, \ldots, p_i$. Note that all pages in this cycle must have the same page rank. To see this assume that there are two pages $p_{i,k}$ and $p_{i,k+1}$ in the cycle that have different page rank. Since $r(p_{i,k+1}) = \text{maximum}\{r(j) \mid j \in B(i)\}$, then $r(p_{i,k+1}) > r(p_{i,k})$. But then by the same argument $r(p_{i,k+2}) \geq r(p_{i,k+1}) > r(p_{i,k})$, $r(p_{i,k+3}) \geq r(p_{i,k+2}) > r(p_{i,k}) \ldots$, $r(p_{i,k-1}) > r(p_{i_k})$, $r(p_{i,k}) > r(p_{i,k})$, which is a contradiction.

      As in a strongly connected graph there is a cycle containing every pair of vertices then by the above argument all pages must have the same rank. Therefore, the page rank exists with this definition. However, since all pages have the same page rank, this definition does not differentiate between low quality and high quality content.

   c. *(10 marks) $r(i) = \frac{1}{n} + \sum_{j \in B(i)} r(j)$.*

      The page rank does not exist in this case because the only way in which the sum of the page ranks is equal to 1 is if each page has page rank $\frac{1}{n}$. However, this is not a solution for the set of equations defined by the above definition of r because since the graph is strongly connected then $B(i) \neq \emptyset$ for any page i; therefore $r(i) = \frac{1}{n} + \sum_{j \in B(i)} r(j) \geq \frac{2}{n}$, which is a contradiction. This is perhaps more easily seen with an example. Suppose that the Web graph is as follows:



This graph is strongly connected. By the definition of page rank, $r(1) = \frac{1}{2} + r(2)$ and $r(2) = \frac{1}{2} + r(1)$. The only way in which $r(1) + r(2) = 1$ is if $r(1) = r(2) = \frac{1}{2}$, but then $r(1) = \frac{1}{2} + r(2) = 1$.

d. *(15 marks)* $r(i) = \text{maximum}\{r(j) \mid j \in B(i)\} + \text{minimum}\{r(j) \mid j \in B(i)\}$, *where* $B(i)$ *is the set of pages that have hyperlinks pointing to page* i. *If* $B(i)$ *has only one page* j, *then* $\text{maximum}\{r(j) \mid j \in B(i)\} = \text{minimum}\{r(j) \mid j \in B(i)\} = r(j)$.

For this definition the page rank does not exist. Since $r(i) = \text{maximum}\{r(j) \mid j \in B(i)\} + \text{minimum}\{r(j) \mid j \in B(i)\} > \text{maximum}\{r(j) \mid j \in B(i)\}$ then by the answer to question (4*b*) the only solution would be for all pages to have the same rank $\frac{1}{n}$. However, if all the pages have the same rank and there is a page i with only two pages j and k pointing to it then $r(i) = r(j) + r(k) = \frac{2}{n}$.

This can be seen also with an example. Consider the same graph from question (4*c*). Then $r(1) = 2r(2)$ and $r(2) = 2r(1)$; but then $r(1) = 2r(2) = 4r(1)$ which implies that $r(1) = 0$ and $r(2) = 0$ and the sum of the page ranks cannot be 1.