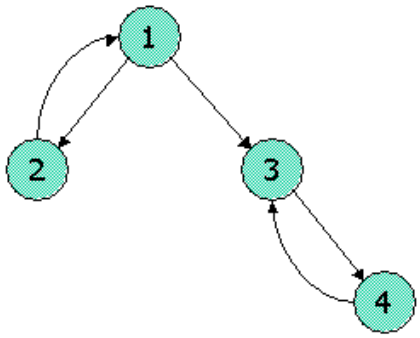
05: PageRank

Number of questions:	5
Positive points per question:	3.0
Negative points per question:	1.0

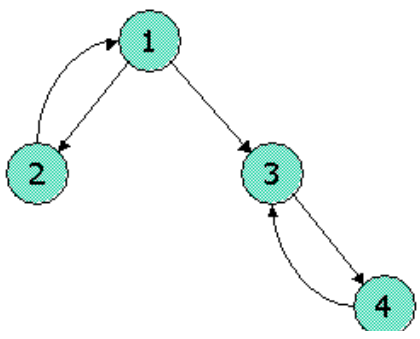
Gradiance quiz on Link Analysis. You can attempt to answer the questions as many times as you like. Questions get randomly regenerated each time. The score of the *last* submission gets saved into our records (that is, once you get a perfect score, don't submit again with a bad one).

1. Compute the Topic-Specific PageRank for the following link topology. Assume that pages selected for the teleport set are nodes 1 and 2 and that in the teleport set, the weight assigned for node 1 is twice that of node 2. Assume further that the teleport probability, (1 - beta), is 0.3. Which of the following statements is correct?



- a) TSPR(1) = .4236
- b) TSPR(4) = .4787
- c) TSPR(2) = .8998
- d) TSPR(3) = .2454

2. Consider the link graph



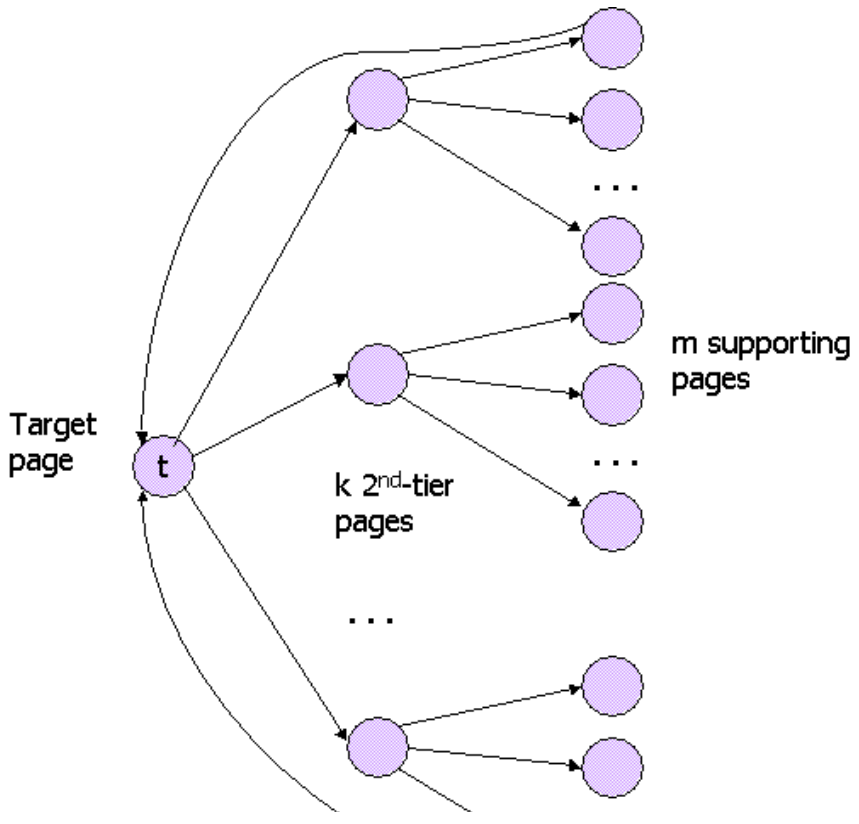
First, construct the L, the link matrix, as discussed in Section 5.5 on the HITS algorithm. Then do the following:

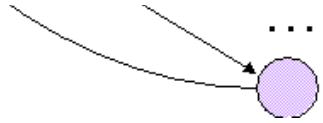
1. Start by assuming the hubbiness of each node is 1; that is, the vector h is (the transpose of) [1,1,1,1].
2. Compute an estimate of the authority vector $\mathbf{a}=L^T\mathbf{h}$.
3. Normalize **a** by dividing all values so the largest value is 1.
4. Compute an estimate of the hubbiness vector $\mathbf{h}=L\mathbf{a}$.
5. Normalize **h** by dividing all values so the largest value is 1.
6. Repeat steps 2-5.

Now, identify in the list below the true statement about the final estimates.

- a) The final estimate of the authority of 2 is 1/8.
- b) The final estimate of the hubbiness of 1 is 3/5.
- c) The final estimate of the authority of 2 is 3/5.
- d) The final estimate of the authority of 4 is 1/8.

3. The spam-farm architecture described in Section 5.4.1 suffers from the problem that the target page has many links --- one to each supporting page. To avoid that problem, the spammer could use the architecture shown below:





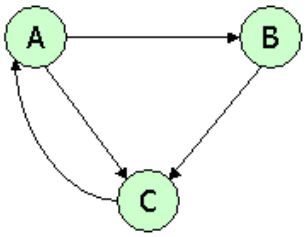
There, k "second-tier" nodes act as intermediaries. The target page t has only to link to the k second-tier pages, and each of those pages links to m/k of the m supporting pages. Each of the supporting pages links only to t (although most of these links are not shown). Suppose the taxation parameter is $\beta = 0.85$, and x is the amount of PageRank supplied from outside to the target page. Let n be the total number of pages in the Web. Finally, let y be the PageRank of target page t. If we compute the formula for y in terms of k, m, and n, we get a formula with the form

$$y = ax + bm/n + ck/n$$

Note: To arrive at this form, it is necessary at the last step to drop a low-order term that is a fraction of 1/n. Determine coefficients a, b, and c, remembering that β is fixed at 0.85. Then, identify the value, correct to two decimal places, for one of these coefficients.

- a) a = 3.07
- b) a = 1.98
- c) b = 0.28
- d) a = 2.59

4. Consider three Web pages with the following links:



Suppose we compute PageRank with β=0.85. Write the equations for the PageRanks *a*, *b*, and *c* of the three pages A, B, and C, respectively. Then, identify in the list below, one of the equations.

- a) c = .9b + .475a
- b) .85b = .575a + .15c
- c) .95a = .9c + .05b
- d) c = b + .575a

5. Consider an implementation of the Block-Stripe Algorithm discussed in Section 5.2 to compute page rank on a graph of N nodes (i.e., Web pages). Suppose each page has, on average, 20 links, and we divide the new rank vector into k blocks (and correspondingly, the matrix M into k stripes). Each stripe of M has one line per "source" web page, in the format:

[source_id, degree, m, dest_1, ..., dest_m]

Notice that we had to add an additional entry, m, to denote the number of destination nodes in this stripe, which of course is no more than the degree of the node. Assume that all entries (scores, degrees, identifiers,...) are encoded using 4 bytes.

There is an additional detail we need to account for, namely, **locality** of links. As a very simple model, assume that we divide web pages into two disjoint sets:

1. **Introvert** pages, which link only to other pages within the same host as themselves.
2. **Extrovert** pages, which have links to pages across several hosts.

Assume a fraction x of pages (0 <= x <= 1) are introverts, and the rest are extroverts. The blocks are arranged such that pages within a host are in the same block. For simplicity, assume that the links from the extrovert pages are spread uniformly across the k stripes (this is reasonably accurate for small values of k).

Construct a formula that counts the amount of I/O per page rank iteration in terms of N, x, and k. The 4-tuples below list combinations of N, k, x, and I/O (in bytes). Pick the correct combination.

Note. There are some additional optimizations one can think of, such as striping the old score vector, encoding introvert and extrovert pages using different schemes, etc. For the purposes of working this problem, assume we don't do any optimizations beyond the block-stripe algorithm discussed in class.

- a) N = 1 billion, k = 3, x = 0.5, 124GB
- b) N = 1 billion, k = 2, x = 0.5, 116GB
- c) N = 1 billion, k = 3, x = 0.5, 132GB
- d) N = 1 billion, k = 2, x = 0.75, 107GB