CIS 833, Fall 2014 **Name:_____**
Exam 2 – 75 minutes

This test consists of four questions. The number of points for each question is shown below.

- Read all questions carefully before starting to answer them.
- Write all your answers on the space provided in the exam paper.
- The order of the questions is arbitrary, so the difficulty may vary from question to question. Don't get stuck by insisting on doing them in order.
- Show your work. Correct answers without justification will not receive full credit. However, also be concise. Excessively verbose answers may be penalized.
- Clearly state any simplifying assumptions you may make when answering a questions.
- **Be sure to write your name on the test paper.**

Question	I	2	3	4	Total
Points	20	20	30	30	100
Your points					

I. Short answer (1-3 sentences) questions [5 questions, 4 points each]

(i) [4 points] Consider the following four text representations of a page – title, body text, URL anchor, and URL text - which can be indexed by a web search engine. For each of the four representations, discuss the main advantage of using it.

- The title is usually a concise and accurate description of a web page. There are few distracting or noisy terms.
- The body text is comprehensive and detailed (although it is also easy to spam).
- The URL anchor is usually a concise description of a web page. Because it is usually assigned by others (i.e., not the author), it is more likely to be objective (i.e., not spammed), and it may be more likely to describe the page as others see it.
- The URL text is concise and may contain terms that are important in navigational queries.

(ii) [4 points] Give two reasons for why it is important for a crawler to detect whether two pages that it has downloaded are "near duplicates".

Near duplicate pages should not be returned as different results on a results page. For the purposes of PageRank, all duplicates and near duplicates should be considered as a single node, e.g. if there are two copies A.1 and A.2, then these should be viewed as a single conceptual node A, where the links into A are the union of the links into A.1 and A.2.

(iii) [4 points] How does the HITS algorithm differ conceptually from the PageRank algorithm?

PageRank computes an authority score for each page, while HITS computes both authorities and hubs on a particular topic. PageRank is useful for answering queries, HITS is best suited for "broad topic" queries rather than for standard page-finding queries (gets a broader slice of common opinion).

(iv) [4 points] Contrast the runtime performance of PageRank with that of HITS.

PageRank is an off-line algorithm - PageRank scores can be precomputed before the query time. HITS is an online algorithm. Authority and hub scores are computed at query time => too expensive in most application scenarios.

(v) [4 points] Suppose that P, Q, and R are different web pages. Explain how it can happen that adding a link from P to Q can raise the PageRank of R. Explain how it can happen that adding a link from P to Q can lower the PageRank of R. In both cases, you should show a specific graph where this happens, although you need not work out the actual numerical values.

First case - initial graph:

P

Q --> R

Adding a link from P to Q raises the PageRank of Q and thus indirectly the PageRank of R. First case:

Second case - initial graph:

P ---> R

Q

If you add a link from P to Q, then P's "contribution of importance" is divided between Q and R rather than going exclusively to R, so the PageRank of R decreases.

II. Binary Independence Model [20 points]

Consider the following document-term frequency matrix, where a 1 entry indicates that the term occurs in a document, and 0 means it does not:

	t1	t2	t3	t4
d1	0	0	1	1
d2	0	1	1	0
d3	0	1	0	1
d4	1	1	0	0

Assume that the number of non-relevant documents is approximated by the size of the collection and that p_i, i.e. the probability of occurrence in relevant documents, is proportional to the probability of occurrence in the collection. Rank the documents in decreasing order of relevance with respect to the queries:

`q1 = {t1, t2}`

`q2 = {t3, t4}`

We need to calculate, RSV for each query/document pairs.

$$RSV = \log \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i(1-r_i)}{r_i(1-p_i)}$$

Given our assumptions, $\log \frac{1-r_i}{r_i}$ can be approximated by $\log \frac{N}{n}$

Also, $\log \frac{p_i}{1-p_i}$ can be approximated by $\log \frac{n}{N-n}$

For query 1 and document 1, we have:

$$RSV = 0$$

For query 1 and document 2, we have:

$$RSV = \log\frac{p_2(1-r_2)}{r_2(1-p_2)} = \log\frac{1-r_2}{r_2} + \log\frac{p_2}{1-p_2} = \log\frac{4}{3} + \log\frac{3}{4-3} = 0.567$$

For query 1 and document 3, we have:

$$RSV = \log\frac{p_2(1-r_2)}{r_2(1-p_2)} = \log\frac{1-r_2}{r_2} + \log\frac{p_2}{1-p_2} = \log\frac{4}{3} + \log\frac{3}{4-3} = 0.567$$

For query 1 and document 4, we have:

$$RSV = \log\frac{p_1(1-r_1)}{r_1(1-p_1)} + \log\frac{p_2(1-r_2)}{r_2(1-p_2)} = \log\frac{4}{1} + \log\frac{1}{4-1} + \log\frac{4}{3} + \log\frac{3}{4-3} = 0.69$$

Thus, the ranking is d4, followed by d2 and d3, followed by d1.

For query 2 and document 1, we have:

$$RSV = \log\frac{p_3(1-r_3)}{r_3(1-p_3)} + \log\frac{p_4(1-r_4)}{r_4(1-p_4)} = \log\frac{4}{2} + \log\frac{2}{4-2} + \log\frac{4}{2} + \log\frac{2}{4-2} = 0.6$$

For query 2 and document 2, we have:

$$RSV = \log\frac{p_3(1-r_3)}{r_3(1-p_3)} = \log\frac{1-r_3}{r_3} + \log\frac{p_3}{1-p_3} = \log\frac{4}{2} + \log\frac{2}{4-2} = 0.3$$

For query 2 and document 3, we have:

$$RSV = \log\frac{p_4(1-r_4)}{r_4(1-p_4)} = \log\frac{1-r_4}{r_4} + \log\frac{p_4}{1-p_4} = \log\frac{4}{2} + \log\frac{2}{4-2} = 0.3$$

For query 2 and document 4, we have:

$$RSV = 0$$

Thus, the ranking is d1, followed by d2 and d3, followed by d4.

III. Query likelihood language model [30 points]

Suppose we have a collection that consists of the 5 documents given in the table below:

DocId	Document
1	dog cat cat
2	dog dog cat cat
3	dog dog dog dog cat cat
4	dog dog cat
5	dog dog dog dog cat

Build the following language models for this collection. For each model, give the ranking of the documents with respect to the query "dog cat cat cat".

(i) Estimate a bigram model of the documents using maximum likelihood estimation (MLE). Remember that the vocabulary of a bigram model consists of all pairs of consecutive words (a.k.a., grams) that appear in the collection.

	Dog cat	Dog dog	Cat cat
D1	1	0	1
D2	1	1	1
D3	1	3	1
D4	1	1	0
D5	1	3	0
Q	1	0	2
C	5	8	3

Q(D1|Q) = (1/2) * (1/2)^2 = 0.125
Q(D2|Q) = (1/3) * (1/3)^2 = 0.037
Q(D3|Q) = (1/5) * (1/5)^2 = 0.008
Q(D4|Q) = (1/2) * (0)^2 = 0
Q(D5|Q) = (1/4) * (0)^2 = 0

(ii) Estimate a bigram model of the documents using smoothed MLE, when smoothing is done by adding 0.5 to the observed counts (remember the renormalization).

	Dog cat	Dog dog	Cat cat
D1	1.5	0.5	1.5
D2	1.5	1.5	1.5
D3	1.5	3.5	1.5
D4	1.5	1.5	0.5
D5	1.5	3.5	0.5
Q	1	0	2
C	5	8	3

Q(D1|Q) = (1.5/3.5) * (1.5/3.5)^2 = 0.078
Q(D2|Q) = (1.5/4.5) * (1.5/4.5)^2 = 0.037
Q(D3|Q) = (1.5/6.5) * (1.5/6.5)^2 = 0.0122
Q(D4|Q) = (1.5/3.5) * (0.5/3.5)^2 = 0.008
Q(D5|Q) = (1.5/5.5) * (0.5/5.5)^2 = 0.0022

(iii) Estimate a bigram model of documents using a mixture model between the documents and the collection with λ=0.3.

$P(W|D) = \lambda(P_{mle}(w|M_d) + (1-\lambda)(P_{mle}(w|M_c)$, where M_c is the language model of the collection.

Considering lambda to be 0.3, we have

Q(D1|Q) = (0.3*(1/2) + 0.7*(5/16)) * (0.3*(1/2) + 0.7*(3/16))^2
Q(D2|Q) = (0.3*(1/3) + 0.7*(5/16)) * (0.3*(1/3) + 0.7*(3/16))^2
Q(D3|Q) = (0.3*(1/5) + 0.7*(5/16)) * (0.3*(1/5) + 0.7*(3/16))^2
Q(D4|Q) = (0.3*(1/2) + 0.7*(5/16)) * (0.3*(0/2) + 0.7*(3/16))^2
Q(D5|Q) = (0.3*(1/4) + 0.7*(5/16)) * (0.3*(0/4) + 0.7*(3/16))^2

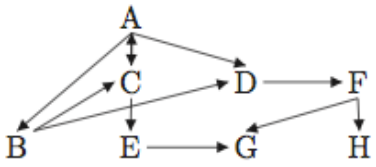
IV. Link Analysis and PageRank [30 points]

(i) Consider the following web graph:

Page A points to pages B, C and D.
Page B points to C and D.
Page C points to A and E.
Page D points to F.
Page E points to G.
Page F points to G and H.

Using page A for starting a web crawl, give the order of the indexing the pages for both breadth-first and depth-first search strategies (with duplicate detection).

The graph looks like this:



The breadth-first order: A, B, C, D, E, F, G, H
The depth-first order: A, B, C, E, G, D, F, H

(ii) Consider the following pages and the set of web pages that they link to:

```
Page A points to pages C, D.
Page B points to page C.
Page C points to pages D, B.
Page D points to pages B.
```

Consider running the PageRank algorithm on this graph of pages. Assume α = 0.15. Simulate the algorithm for two iterations. Show the page rank scores for each page twice for each iteration, both before and after normalization.

R = [0.25, 0.25, 0.25, 0.25]
E = [0.0375, 0.0375, 0.0375, 0.0375]

Iteration 1:
R' = [0.0375, 0.356, 0.356, 0.25]
Norm R = [0.0375, 0.356, 0.356, 0.25]

Iteration 2:
R' = [0.0375, 0.4013, 0.356, 0.2047]
Norm R = [0.0375, 0.4013, 0.356, 0.2047]

(iii) Consider running the HITS (Hubs and Authorities) algorithm on the same graph of pages as in (ii), shown again here:

```
Page A points to pages C, D.
Page B points to page C.
Page C points to pages D, B.
Page D points to pages B.
```

Simulate the algorithm for two iterations. Show the authority and hub scores for each page twice for each iteration, both before and after normalization.

Iteration1:

Before Normalization:

a (A) = 0 a(B)=2 a(C) =2 a(D) =2

h(A) =2 h(B) =1 h(C) =2 h(D) =1

c(a) = sqrt (0+4+4+4) = 3.464

c(h) = sqrt (4+1+4+1) = 3.16

After Normalization:

a(A) = 0 a(B)=0.577 a(C) =0.577 a(D) =0.577

h(A) =0.63 h(B) =0.31 h(C) =0.63 h(D) =0.31

Iteration2:

Before Normalization:

a (A) = 0 a(B)=0.94 a(C) =0.94 a(D) =1.26

h(A) =1.54 h(B) =0.577 h(C) =1.54 h(D) =0.577

c(a) = sqrt (0+0.94^2+0.94^2+1.26^2) = 1.831

c(h) = sqrt (1.54^2+0.577^2+1.54^2+0.577^2) = 2.32

After Normalization:

a(A) = 0 a(B)=0.513 a(C) =0.513 a(D) =0.688

h(A) =0.66 h(B) =0.248 h(C) =0.66 h(D) =0.248