Scores:

1)	5)
2)	6)
3)	7)
4)	8)

COMP 4321 Search Engine for Web and Enterprise Data
Mid-term Examination, Spring 2021
April 8, 2021
Time Allowed: 1 hr 15 min

Name: ______________________________ Student ID: ________________________

Note: Note: You can allocate enough space for each question and hand-write or type your answers into the file. Answers must be precise and to the point.

1. **[5]** About link-based ranking methods:
   (a) HyPursuit uses links to infer the similarity between pages and use page similarities to group/cluster similar pages in the same groups.
   (b) Wise uses links to pass similarity scores from the parent pages to the child pages and use the scores to rank pages.
   (c) Google uses links to infer the authority or quality of a page.
   (d) PageRank and Hypursuit are query independent
   (e) Page similarity scores in Wise are query dependent

   Circle the correct choice:
   (i) All of the above are true
   (ii) Only (a), (b), (c) are true
   (iii) Only (b) and (c) are true
   (iv) Only (c) is true

2. [5] A web graph contains only four pages. Page A points (links) to Pages B, C and D and there is no other links. Now, a new page X is created and points (links) to B.
   (a) The hub weights of A, B, C and D will increase
   (b) The authority weights of A, B, C and D will increase
   (c) The hub weights of B, C and D will increase
   (d) The authority weights of B, C and D will increase
   (e) The hub weight of A will increase

   Circle the correct choice:
   (i) All are correct
   (ii) Only (b), (c) and (e) are correct
   (iii) Only (c) and (d) are correct
   (iv) Only (d) and (e) are correct

3. **[5]** A spider/crawler need to face the following challenges:

T [✓ F] a) It has to get approval from website owners before it can crawl their websites
[✓ T] F b) It has to deal with network timeouts, server timeouts and no-responses
T [✓ F] c) It is difficult the find out the page URLs to crawl
[✓ T] F d) It takes a lot of time and computing resources to crawl a large number of webpages
T [✓ F] e) Every website welcomes frequent visits from spiders/crawlers

Circle the correct choice:
(i) All of the above
(ii) None of the above
(iii) (a) and (b) only

(iv) (b) and (d) only

4. **[8]** Which of the following statement(s) is/are correct about the vector space model?

[✓ T] F a) Terms are assumed to be independent in the document collection
[✓ T] F b) Retrieval resembles Boolean OR (disjunction) on the query terms
T [✓ F] c) Term weights must be based on tf and idf
T [✓ F] d) Retrieval guarantees pages matching the largest number of query terms receive top ranks
[✓ T] F e) Vector space to documents is geographic space to objects on earth

Circle the correct choice:
(i) All of the above
(ii) (a), (b) and (e) only
(iii) (a) and (b) only
(iv) (b) and (d) only

5. **[10]** Give two advantages and two disadvantages of the ways that Clever incorporates HITS into a search engine compared to Google which is based on precomputation of PageRank.

I give three below; any two are fine.

Advantages:
(i) Hubs and Authorities are computed based on the query and thus is suitable for search engine
(ii) Hubs and authorities are computed based on a subgraph that are relevant to (the topic of) the query.
(iii) HITS gives Hub weights, which could be useful for search engine but are not available in PR.

Disadvantages:
(i) Inefficient because Hubs and Authorities are computed during query processing time
(ii) The subgraph for computing Hubs and Authorities are small so that the weights are not representative.
(iii) It is hard to control the size of the topical subgraph; too large would give you loose topics while too small gives you unreliable hub/authority weights.

There could be other advantages and disadvantage, and I am flexible with it as long as it indicates an understanding of the methods. E.g., Clever could be easily built on top of an existing search engine but on the other hand a bad search engine could lead to bad search result in Clever.

6. **[20]** Given the inverted file structure used in the lecture (i.e., an index of sorted index terms and lists of posting lists sorted by document ID):

(a) [5] When the number of documents increases to billions, processing speeds for both Boolean and Vector Space models will degrade. What is the major cause of the speed degrade?

When the number of documents increases to billions, the postings lists would be extremely long. Thus, processing speed slows down significantly because long postings list have to be retrieved and processed.

[Just a note: Since the number of indexed terms will in general saturate and the indexed terms are stored in a b-tree or hash file, processing speeds will not be degraded significantly as compared to the processing of postings lists.]

(b) [5] Suggest one way to alleviate the slowdown due to the above scalability problem.

Multiple indexes have to be used so that the postings lists in an index are short.

Another way is to avoid using a consecutive list for the postings lists, e.g., by using a b-tree. This method does not scale to billions of pages but I will accept it as correct.

(c) [5] In cosine similarity measure, why is the normalization factor expensive to compute if we use tf*idf as term weight?

In the vector space model, document length normalization is to divide the document score of a document (typically obtained from inner product computation) by the vector length of the document.

Computation cost is extremely high since it requires the computation of the weights of all document terms, which are numerous, and the weights cannot be precomputed because of dynamically changing idf values.

(d) [5] Explain why the inverted file alone cannot support document deletion. Give a brief description of the additional file structure needed to support deletion efficiently.

The inverted file is able to find documents given keywords. When a document is deleted, we only know the document ID. Thus, we need to find the keywords in a document, which the inverted file cannot handle. To resolve the problem, we need to keep a forward index mapping document ID to keywords in that document.

7. [15] Given the following two documents vectors, showing the terms and the terms' <u>weights</u> in the documents:
D1: 〈 database 1.0, dataset 1.0 〉
D2: 〈 consumer 1.0, dataset 1.0 〉

(i) [5] For the vector space query Q = 〈consumer, dataset〉, compute the cosine similarity between Q and the documents. Assume query term weights are equal to 1. Show the computation of the L2 magnitudes of Q, D1, D2, the inner products between Q and D1, and Q and D2, and finally, the cosine similarities of Q and D1, and Q and D2.

||Q|| = sqrt (1**2 + 1**2) = 1.41
||D1|| = sqrt (1**2 + 1**2) = 1.41
||D2|| = sqrt (1**2 + 1**2) = 1.41
Q•D1 = 0 + 1 = 1
Q•D2 = 1 + 1 = 2
Sim(Q, D1) = 1 / (1.41*1.41) = 0.5
Sim(Q, D2) = 2 / (1.41*1.41) = 1

(ii) [5] Compute the centroid of the documents.

The document vectors are:

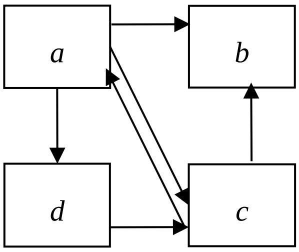
	consumer	database	dataset
D1	0	1.0	1.0
D2	1.0	0	1.0
Centroid	0.5	0.5	1.0

(iii) [5] (This question is independent of (i) and (ii)) Assuming that there are many document groups/clusters, describe a method to use the centroid to speed up the ranking process in vector space model and highlight if there is any sacrifice in your method.

The general idea is to compute the similarity of the query vector and the centroid vectors of the clusters, and filter away clusters that have low similarity. Then, compute the similarity of the query vector against the document vectors in the remaining clusters and return the results. The sacrifice is that some eliminated clusters may contain some highly relevant results, and these results will be lost.

[I suppose the students can come up with ingenious answers.]

8. **[15]** Compute the PageRank, Hub and Authority weights of the pages in the following graph.



(a) [4] Write down the PageRank, Authority and Hub formulas:

PageRank
PR(a) = (1-d) + d(PR(c) / 2)
PR(b) = (1 - d) + d(PR(a) / 3 + PR(c) / 2)
PR(c) = (1 - d) + d(PR(a) / 3 + PR(d))
PR(d) = (1 - d) + d(PR(a) / 3)

Hubs
Hub(a) = Aut(b) + Aut(c) + Aut(d)
Hub(b) = 0
Hub(c) = Aut(a) + Aut(b)
Hub(d) = Aut(c)

Authority
Aut(a) = Hub(c)
Aut(b) = Hub(a) + Hub(c)
Aut(c) = Hub(a) + Hub(d)
Aut(d) = Hub(a)

(b) [6] Compute by hand (no programming is required) the PageRank, Authority and Hub weights in the following table. Use L1 norm to normalize the weights in each iteration. In each table cell, write down the weights before and after normalization and show intermediate computational steps (make sure writing is readable). For PageRank, set damping factor d=1.

PageRank:

	0	Iteration 1	Iteration 2
a	1/4	1/8 1/6	2/9 4/13

b	1/4	5/24 0.2083 5/18	5/18 0.307692 5/13 0.384615
c	1/4	1/3 4/9	1/6 0.16666 3/13 0.230769
d	1/4	1/12 0.0833 1/9 0.1111	1/18 0.055556 1/13 0.076923

Authority:

	0	**Iteration 1**	**Iteration 2**
a	1/4	1/4 1/6	1/3 1/7
b	1/4	1/2 1/3	5/6 5/14 0.357
c	1/4	1/2 1/3	2/3 2/7 0.286
d	1/4	1/4 1/6	1/2 3/14 0.214

Hub:

	0	**Iteration 1**	**Iteration 2**
a	1/4	3/4 1/2	5/6 1/2
b	1/4	0	0
c	1/4	1/2 1/3	1/2 3/10
d	1/4	1/4 1/6	1/3 1/5

(c) [5] Based on reasoning on the graph connections, briefly discuss the convergence trend of the three weights without actually computing the weights for a large number of iterations.

d = 1 means no teleporting. In this case, since Node b is a dangling node, PR computation would not converge. Hub and Authority converge since dangling nodes do not cause problem with HITS.

9. **[15] (a) [5]** What is the definition of fallout ratio? What problems does it solve that precision and recall cannot?

Fallout = number of non-relevant documents retrieved / total number of non-relevant documents

Problems solved: practically no zero division; useful to judge performance of queries which have lot of relevant results.

(b) [10] The following table shows the ranked search results of a query. A 0 score means non-relevant, 1 relevant, 2, very relevant, 3, strongly relevant. Give the ideal ranking of the 5 documents. Then, compute DCG_5, compute $IDCG_5$, and $NDCG_5$.

Doc ID	**Rank**	**Score**
A	1	1
B	2	0
C	3	2
D	4	3
E	5	0

Ideal ranking: D, C A, followed by B and E or E and B.
DCG5 = 1/log2(2) +2/ log2(4) +3/ log2(5) = 3.29
IDCG5 = 3/log2(2) + 2/log2(3) + 1/log2(4) = 4.76
NDCG5 = 3.29/4.76 = 0.69