Midterm 1
Date: March 6, 2017 (in class)
Time: 4:00 ~ 5:15 PM (75 minutes)
Total: 20 points
8 points: 16 True/False questions: 0.5 points each
12 points: 4 groups of questions with 5-6 sub questions per group

20% of the total score will be directly related to the quizzes.

Closed book, NO computer, No cheat sheets

A. Topics covered
1. Distributed File Systems
   - Architecture of GFS
   - Master operations
   - Prefix compression
   - Scalability of GFS
   - Read/write operations
   - Relaxed consistency
   - File state regions
   - Data flow of the “write” operation
   - Atomic record append
   - Locks and snapshot
   - Garbage collection
   - CAP theorem

2. MapReduce
   - Job submission, execution, and monitoring
   - Fault tolerance
   - Shuffle and Sort
   - Combiner
   - Split and partition

3. Text Mining: Finding similar Items at Scale
   - Jaccard Similarity
   - Cosine Similarity
   - TF-IDF
   - Using n gram
   - NN search
   - Minhashing and permuting
   - Locality sensitive hashing
4. Link Analyses
   • Inverted index
   • Regular PageRank algorithm
   • Matrix multiplication with MapReduce
   • PageRank algorithm with Taxation
   • Calculation PageRank algorithm with Dead ends
   • Using MapReduce to calculate PageRank values
   • Understanding link farm and link spam
   • Spam mass and TrustRank

5. Filtering Patterns
   • Random filter
   • Bloom filter
B. Sample “Group Problems”

Sample Question A.
Suppose that we analyze data for an online music download service. The rows of this table are the songs listed in the service. The columns are customers, represented by the set of products they bought.

<table>
<thead>
<tr>
<th>Row Number (r)</th>
<th>Element</th>
<th>Ann</th>
<th>Tom</th>
<th>Bob</th>
<th>Joe</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Song A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>Song B</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Song C</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Song D</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Song E</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Question 1. What is the minhash value of the set for Ann?
  a. 0   b. 0, 2   c. 0, 2, 3   d. 2   e. 4

Question 2. What is the minhash value of the set for Bob?
  a. 1   b. 1,3   c. 1,3,4   d. 3   e. 4

Suppose that we use a hash function $h$ to permute the list of songs. The result is,

<table>
<thead>
<tr>
<th>$h(r)$</th>
<th>Element</th>
<th>Ann</th>
<th>Tom</th>
<th>Bob</th>
<th>Joe</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Song C (index 0)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>Song E (index 1)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Song B (index 2)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>Song A (index 3)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Song D (index 4)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Question 3. What is the minhash value of the set for Ann?
  a. 0   b. 1   c. 1,2   d. 1,3   d. 1,3,4

Question 4. What is the minhash value of the set for Bob?
  b. 1,2   c. 2,4   d. 1,2,4   e. 4
Suppose that we generate Minhash Signatures using hash functions $h_1, h_2, h_3,$ and $h_4$.

- $h_1(x) = (x + 1) \mod 5$
- $h_2(x) = 2x \mod 5$
- $h_3(x) = (3x + 1) \mod 5$
- $h_4(x) = x \mod 5$

### Table 4-b. Intermediate values

<table>
<thead>
<tr>
<th>Row Number</th>
<th>Element</th>
<th>Ann</th>
<th>Tom</th>
<th>Bob</th>
<th>Joe</th>
<th>$h_1(r_{number})$</th>
<th>$h_2(r_{number})$</th>
<th>$h_3(r_{number})$</th>
<th>$h_4(r_{number})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Song C (index 0)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>Song E (index 1)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Song B (index 2)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Song A (index 3)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Song D (index 4)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

### Table 4-c. Initial table to generate Minhash signatures

<table>
<thead>
<tr>
<th>Hash functions</th>
<th>Ann</th>
<th>Tom</th>
<th>Bob</th>
<th>Joe</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$h_2$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$h_3$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$h_4$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
</tbody>
</table>

**Question 4-3.** What is the Minhash Signature $<h_1, h_2, h_3, h_4>$ of the set for Ann?

- a. $<2, 2, 4, 1>$
- b. $<0, 1, 2, 1>$
- c. $<1, 0, 1, 0>$
- d. $<0, 2, 2, 1>$
- e. $<0, 0, 1, 0>$
Sample Question B.

The diagram above depicts a set of chunk servers in the Google file system. Suppose that there have been 3 operations processed sequentially in the order listed below.

OPERATION 1: Data sub-block 1 (size of 16MB) is stored at the Primary chunk server and also the Replica chunk servers A and B successfully at offset 0.

OPERATION 2: Data sub-block 2 (size of 16MB) is stored at the Primary chunk server and the Replica chunk server A and B successfully at offset k.

OPERATION 3: Data sub-block 3 (size of 16MB) is stored at the Primary chunk server successfully at offset m. The replica chunk server A failed in the middle of the storage operation for sub-block 3 at the offset n. The replica chunk server B failed in the storage of sub-block 3 at offset p.

The client tried to send this data sub-block 3 again after it received an error message from the primary chunk server (OPERATION 4).

1) Will the primary chunk server store the new data sub-block 3 again? (Yes/No)
   If it does, what will be the offset for the start of the new data sub-block 3 in the primary chunk server? To specify the offset, you can use the offsets defined above and any integer numbers.
   a. m   b. m+n   c. m+k   d. m+p

2) Will the replica chunk server A store the new data sub-block 3 again? (Yes/No)
   If it does, what will be the offset for the start of the new data sub-block 3 in the replica chunk server A? To specify the offset, you can use the offsets defined above and any integer numbers.
   a. m   b. m+n   c. m+k   d. m+p
3) Will the replica chunk server B store the new data block 3 again? (Yes/No)  
   If it does, what will be the offset for the start of the new data subblock 3 in the replica 
   chunk server B? To specify the offset, you can use the offsets defined above and any 
   integer numbers. 
   a. m      b. m+n      c. m+k      d. m+p

**Sample Question C.**

Suppose that a snapshot operation is performed for a directory, “/project/googlemap/coloradousers”. New directory is “/backup/googlemap/coloradousers”. Design a write lock and read lock for the snapshot operation in GFS to prevent creating a new file “/project/googlemap/coloradousers/CSUusers”, while this snapshot operation is being processed.
Sample Question D.

Suppose that 10 items are registered to the initial Bloomfilter B using the same set of hash functions. Assume that the current bloomfilter has the state depicted below.

\[ h_1(x) = x \mod 15 \]
\[ h_2(x) = (x+3) \mod 15 \]
\[ h_3(x) = (x+2) \mod 15 \]

<table>
<thead>
<tr>
<th></th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>09</th>
<th>08</th>
<th>07</th>
<th>06</th>
<th>05</th>
<th>04</th>
<th>03</th>
<th>02</th>
<th>01</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

(A) The integer 4 is not a member of the set. (True/False)
(B) The integer 9 may be a member of the set. (True/False)
(C) The integer 1 must be a member of the set. (True/False)

Sample Question E.

Consider that you are calculating PageRank values for web pages. There are 10 Billion web pages and you have created a 10 Billion x 10 Billion transition matrix \( M \). As a part of iterative computations, you use the MapReduce computing framework without Taxation. The \( k^{th} \) iteration of the MapReduce job will create a vector \( v^{(k)} \) with 10 Billion items. The \( j^{th} \) item in \( v^{(k)} \) is calculated using the following formula:

\[ v_j^{(k+1)} = \sum_j m_{ij} v_j^{(k)} \]

**Question 1.** What are the values \( m_{ij} \) stored in the transition matrix \( M \)?

a. The total number of times that web page \( i \) has been visited
b. The probability that page \( i \) is to be visited from the \( j^{th} \) page
c. The page \( i \)'s page rank value after \( j^{th} \) iteration
d. Random number generated by server

**Question 2.** What are the values \( v_j \) for the \( k^{th} \) iteration?

a. The average PageRank value of the page \( j \) after the \( k^{th} \) step
b. The probability that the surfer was at the node \( j \) at the \((k-1)^{th}\) step
c. The highest PageRank value of the page \( j \) after the \( k^{th} \) step
d. The lowest PageRank value of the page \( j \) after the \( k^{th} \) step