NATIONAL UNIVERSITY OF SINGAPORE

|  |
| --- |
| SCHOOL OF COMPUTING  ASSESSMENT FOR  Semester 1 AY2021/2022  CS5425/CS4225 – Big Data Systems for Data Science  Final Test Paper  Nov 11 2021 Time Allowed: 1.5 Hours |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INSTRUCTIONS TO STUDENTS:**

1. This assessment paper contains **FOUR (4)** questions and comprises **THIRTEEN (13)** printed pages, including this page.
2. Students are required to answer ALL the questions.
3. Write your answers within the space provided. Answers written on other parts of the answer script will not be graded unless you specify explicitly.
4. This is an **OPEN BOOK** examination, allowing any materials except discussion with others.
5. Please circle your class and write your matriculation number below.

**Class: CS5425 CS4225**

**Matriculation Number: A0184679H**

This portion is for examiner’s use only

|  |  |  |
| --- | --- | --- |
| **Question** | **Marks** | **Remarks** |
| **Q1 [9 marks]** |  |  |
| **Q2 [13 marks]** |  |  |
| **Q3 [8 marks]** |  |  |
| **Q4 [10 marks]** |  |  |
| **Total [40 marks]** |  |  |

**QUESTION 1: True / False and Computation Questions. [9 marks]**

**A brief justification for each answer is sufficient. Each question is worth 1 mark.**

1. Assume that we are using MinHash for detecting duplicate documents as described in lecture. If two documents have the same MinHash signature, they must be exact duplicates.

**Answer: False**

**Explanation: Two documents with same minHash indicates that the documents have a k-shingle which causes the hash value to be minimum, and the other shingles in the document can be different, which is not an exact duplicate.**

1. Assume that we are using MinHash for detecting duplicate documents as described in lecture. If two documents are exact duplicates, they must have the same MinHash signature.

**Answer: True**

**Explanation: If two documents are duplicates, their k-shingle which cause the hash value to be a minimum is the same, therefore the MinHash must be the same.**

1. In Spark, some functions including groupByKey() and join() require the data in the RDDs we are processing to pass through the master node, and hence can cause significant slowdown.

**Answer: True**

**Explanation: These functions are bottlenecked by the shuffling step when returning the results to the master node, but only when an action is called.**

1. Let G be an undirected graph. In the PageRank algorithm, the problem associated with spider traps (as discussed in lecture) cannot occur in this case.

**Answer: True**

**Explanation: A spider trap exists only when there is a group with all out links within the group. When the graph is undirected, the direction is not enforced therefore there is no loop.**

1. In PageRank, self-loops cause the same problem as dead-ends as discussed in lecture, and should be handled by preprocessing the random walk matrix () in the same way as for dead-ends.

**Answer: True**

**Explanation: Can only be solved by teleportation to avoid pagerank leaking out.s**

1. In topic-sensitive PageRank, the node with the highest topic-sensitive PageRank score is always in the teleport set.

**Answer: False**

**Explanation: The topic-sensitive PageRank is also dependent on the ß value. In lecture 9 slide 62, node 3 has highest score but it is not in the teleport set.**

1. In stream processing systems, if the length of the stream keeps increasing without bound, the stream processing system will eventually run out of memory.

**Answer: False**

**Explanation: With algorithms like reservoir sampling, we can make sure that each element has same probability of being selected to store it a fixed size storage, so it will not run out of memory because memory size is fixed.**

1. You are tasked to evaluate a newly developed diagnostic test (or alternately: a new machine learning algorithm). You test it on 100 cases which you know to be positive, and the test returns positive results on 90 of them. On 50 cases known to be negative, the test returns positive results on 20 of them.   
   Based on these results, what is the number of false negatives?

**Answer (explanation not required): 10**

1. Based on the same results as the previous question, what is the specificity of the test?

**Answer (explanation not required; and you can answer as a fraction): 30 / 50**

**QUESTION 2: MapReduce and Spark [13 marks]**

(a) On a certain large Spark cluster, you find that running jobs with the sortBy() transformation (i.e. sort the data) or collect() on an RDD cause the Spark job to run very slowly, but the filter() transformation is much faster. Provide a brief explanation based on a property of each of the above operations:

[3 marks]

1. sortBy():

**Answer: The bottleneck of the sortBy operation is in the shuffling step. In sorting, it will take almost O(n log n) time complexity, which is slow when RDD is huge.**

1. collect():

**Answer: collect() is an action, therefore running collect() will execute all the transformations on the RDD since transformations are lazy, and will only run when an action is called, and collect() also retrieve elements from RDD to driver node.**

1. filter():

**Answer: filter() is a transformation, therefore it is lazy as it will not execute until an action is called. Filtering also takes O(n), where n is the size of RDD, therefore it is faster than sorting.**

(b) You find that the same phenomenon described in part (a) still happens even when the size of the RDD becomes very small (e.g. an RDD containing 1000 integers). Explain why this is the case.

[2 marks]

**Answer: The machine on the Spark cluster could be busy running other tasks.**

(c) Suppose you want to run the following Spark program to process a massive log file, on a large cluster. However, the program crashed. Explain why this is the case, and which line(s) you would modify to fix the problem.

[2 marks]

Line 1: info = spark.textFile(“hdfs://log”)

Line 2: info = info.map(lambda s: s.split(“ ”)[2])

Line 3: info.collect()

Line 4: info.filter(lambda s: “hadoop” in s).count()

Line 5: info.filter(lambda s: “spark” in s).count()

**Answer: Line 3 causes the program to crash as the log file is huge. Should use info.cache() to save the RDD to memory to avoid expensive computation.**

(d) Suppose you have a large number of text documents, and want to compute the MinHash signature of each document based on its *k*-shingles, where ***k*=1**. We read our documents line by line, receiving input key-value pairs of the form <*docID*, *line*>, where *docID* is the document ID being read, and *line* is a **space-separated string** containing a line of the document. Show pseudo-code for how you would use MapReduce **with combiners (**not in-mapper combiner) to compute the MinHash value for each document, using a single hash function (which maps strings to integers); namely hash(x). Follow the pseudo-code of the below format. Assume that we cannot guarantee that the input lines will come in any particular order. You can assume that the input (in *line*) is ‘clean’; e.g. no duplicate spaces or spaces at the start / end of the line, and no characters other than letters and spaces are present. You can assume the existence of a string splitting function of your choice, e.g. split().

map (docID, line) {  
/\* your pseudo code\*/  
/\* you need to output the map results by calling the API, emitIntermediate(specify your map output)\*/  
}

reduce (/\* specify your input to reducer \*/) {  
/\* your pseudo code \*/  
/\* you need to output the map results by calling the API, emitFinal(docID, minHash). Here, the key should be document ID and the value should be its MinHash value \*/  
}

combine (/\* specify your input to combiner \*/) {  
/\* your pseudo code \*/  
/\* emit appropriate outputs for the combiner to work \*/  
}

[6 marks]

**Answer:**

map (docID, line) {

line = line.split(“ “); // split by space

for word in line {

hashValue = hash(word);

emit(docID, hashValue);

}  
}

reduce (docID, hashes) {

minHash = INFINITY;

for hashValue in hashes {

minHash = min(minHash, hashValue); // compare  
}

emit(docID, minHash); // final minHash

}

combine(docID, hashes) {

for hashValue in hashes {

emit(docID, hashValue); // combiner do nothing  
}

}

**QUESTION 3: Graphs [8 marks]**

(a) Billy wants to compute the power iteration algorithm for PageRank in Pregel / Giraph, except that he wants to modify the stopping condition: the algorithm should stop once ***every*** vertex’s PageRank value changes by less than 0.01 in one step; that is, letting be the rank of node *i* at superstep *t,* we want the algorithm to stop once , for all nodes *i* in the graph.

Billy says that this stopping condition can be implemented by making the compute() function for each vertex call voteToHalt() whenever the difference between the vertex’s current and previous value is less than 0.01 (and aside from the stopping condition, the rest of the implementation is as we have seen in lecture). Will this compute the correct result or not, and why?

[4 marks]

**Answer: No, this will get the wrong result. For example, a vertex with no incoming edges but only has outgoing edges, its value remain the same but other vertices that receive PageRank from this vertex will change. Calling voteToHalt on this vertex will not work, as other pages PageRank will still be updated in the next iteration.**

(b) Assume that we use topic-sensitive PageRank to obtain scores for webpages for the “data science” topic. There is a set of webpages, none of which are in the “data science” seed set (i.e. teleport set) . Assume that for each page in , if it has any incoming links, these links can only come from other pages in . Is it true that all pages in have a topic-sensitive PageRank score of 0? Explain why or why not. (You do not need to give a formal proof; a clear enough explanation is sufficient).

[4 marks]

**Answer: False. If a node has incoming link, it will never have a PageRank of 0, by the matrix formulation, A = beta \* M + (1 – beta) \* N. Even if N is 0, the incoming edges will still contribute to the PageRank score.**

**QUESTION 4: Application and Problem Solving [10 marks]**

(a) Assume that your e-commerce company’s webpage has a NoSQL document store database containing data about users visiting the webpage (e.g. IP address, country, time spent browsing the webpage, number of products they have bought, etc.) Your CEO suggests using the Range Partitioning scheme with the number of products they have bought as a partition key. State and explain a possible benefit of this choice, and a possible disadvantage of this choice.

[4 marks]

**Answer: Possible benefit is able to answer range-based queries. For example, want to find out customers who bought between 100 products to 200 products on 11/11 Shopping Day. Possible disadvantage is the data partitioned could be imbalanced, causing heavy workload on one data shard.**

(b) Assume that your company processes a massive number of page views every second, each tagged with information about which product type was viewed, the time the user spends browsing a page, etc. Your CEO has tasked you to design a flexible approach that can help provide an up-to-date answer at any time for a **wide variety** of possible queries about this data (e.g. which products are most popular; average time spent browsing each product, etc.) Your program should give an answer efficiently no matter how large the data is, but is allowed to be slightly inaccurate. Which ***algorithm*** discussed in class is suitable, and why? Explain how you would use this algorithm.

[4 marks]

**Answer: The reservoir sampling algorithm discussed in class can be used. With a reservoir of fixed sized, the program is able to answer queries like most popular products by selecting the max product from the reservoir. For other queries, can treat the reservoir as the whole stream and extract each statistics as each item is uniformly being sampled, and little bugs are allowed.**

(c) State (and explain) which of the ***systems*** covered in lecture is suitable for satisfying the CEO’s goal, and explain why it is suitable.

[2 marks]

**Answer: Storm. Since the number of page views is processed each second, this is a streaming problem, can use Storm as a stream data processing system.**

BLANK PAGE

END OF ASSESSMENT