LEAVE THIS SPACE EMPTY

DO NOT MOVE THE QR CODE BOX IN THE BOTTOM RIGHT CORNER

KEEP THE SPACE AROUND THE QR CODE BOX FREE

DELETE THIS TEXT BOX

UNSW Sydney

Term 3, 2024 Final Examination

COMP9313:

Instructions:

- Time allowed – 2 hours
- Reading Time – 15 minutes
- This paper contains 6 questions on the 11 following pages.
- Total marks available – 50, worth 50% of the total marks for the course.
- Marks available for each question are shown in the exam.
- Double-check that you have filled out your zID (written and bubble-filled correctly)
- Students are advised to read all examination questions before attempting to answer the questions.
- All answers must be written in ink. Except where they are expressly required, pencils may be used only for drawing, sketching or graphical work.
- Any assumptions found to be necessary in answering the questions should be stated explicitly.
- Show all your workings to receive any partial marks applicable.
- Do not write or draw on any part of the QR codes on the question booklet.
- Extra pages have been provided at the back if more working space is needed. If you use these pages, be sure to label them with the question number.
- This paper may **NOT** be retained by the candidate.
- Students may bring to the examination – printed or hand written notes on one A4 page (double sided)

Notes:

Question 1. Concepts (6 marks)

(i) (2 marks) Explain the data flow in MapReduce using the word count problem as an example.

You need to mention the following points at least:

1. The mapper read the data from HDFS. A mapper task is started for each HDFS data block. Each record is processed by one map function. The mapper output is sorted and stored on local disks.
2. In order to reduce the mapper output size and save the network I/O cost, a combiner can be designed to perform local aggregation.
3. According to the number of reducers specified, the mapper output is partitioned to several groups, each corresponding to one reducer.
4. The reducer fetches its partition from each mapper and merges the locally sorted data to obtain a global order. Next, each key and its associated values are processed by one reducer function. The output is stored back to HDFS.

(ii) (2 marks) Explain the data flow in Spark using the word count problem as an example.

You need to mention the following points at least:

1. RDD/DataFrame operations are categorized into transformation and action, and the transformation operations are lazily evaluated.
2. When reaching an action, a job will be created, and the real execution begins at this point.
3. Each transformation operation that requires data shuffling becomes the boundary of stages. Spark create a DAG based on the stages.
4. Within each stage, a task will be run for each partition on an executor, and the tasks run in parallel.

(iii) (2 marks) In Project 2, what is the best data structure to store the high-frequency words when broadcasting them? Explain the reasons.

Set. In Python, the search complexity using a set (expected to be O(1)) is better than that of using a list or an array (log(N)).

Question 2. MapReduce Programming (10 marks)

Requirement: *You should explain how the input is mapped into (key, value) pairs by the map stage, i.e., specify what is the key and what is the associated value in each pair, and how the key(s) and value(s) are computed. Then you should explain how the (key, value) pairs produced by the map stage are processed by the reduce stage to get the final answer(s). You only need to provide the pseudo code for Mapper and Reducer (optionally Combiner and configuration for Partitioner, if necessary). The* ***efficiency*** *of your method will be considered.*

(i) (4 marks) Given a table shown as below, find out the person(s) with the maximum salary in each department (employees could have the same salary).

EmployeeID	Name	DepartmentID	Salary
001	Emma	1	100,000
002	Helen	2	85,000
003	Jack	3	85,000
004	James	1	110,000

(Concise version)

Mapper: for each record, Emit(department + “,” + salary, name)

Combiner: find out all persons with the local maximum salary for each department

Reducer: receives data ordered by (department, salary), the first one is the maximum salary in a department. Check the next one until reaching a smaller salary and ignore all remaining. Save all persons with this maximum salary in the department

JOBCONF: key partitioned by “-k1,1”, sorted by “-k1,1 -k2,2nr”

(ii) (6 marks) Assume that you are given a data set crawled from a location-based social network, in which each line of the data is in format of (userID, a list of locations the user has visited <loc1, loc2, …>). Your task is to compute for each location the set of users who have visited it, and the users are sorted in ascending order according to their IDs.

(Detailed version)

```
class Question1

    method map(self, userID, list of locations)
        foreach loc in the list of locations
            Emit("loc, userID", "")

    method reduce_init(self)
        current_loc = ""
        current_list = []

    method reduce(self, key, value)
        loc, userID = key.split(",")
        if loc != current_loc
            if current_loc!=""
                Emit(current_loc, current_list)
            current_list = []
            current_list.add(userID)
            current_loc=loc
        else
            current_list.add(userID)

    method reduce_final(self)
        Emit(current_loc, current_list)

In JOBCONF, configure:
        'mapreduce.map.output.key.field.separator':',',
        'mapreduce.partition.keypartitioner.options':'-k1,1',
        'mapreduce.partition.keycomparator.options':'-k1,1 -k2,2'
```

Question 3. Spark Programming (10 marks)

Provide the PySpark code for the given problems (minor errors are acceptable).

(a) (5 marks) **RDD programming (DataFrame APIs not allowed):** (5 marks) **RDD programming (DataFrame APIs not allowed):** Given a set of marks from different courses (the input format is as shown in the left column), the task is to compute average marks for every course and sort the result by course_name in alphabetical order.

Input:	Output:
student1:course1,90;course2,92;course3,80;course4,79;course5,93 student2:course1,92;course2,77;course5,85 student3:course3,64;course4,97;course5,82	course1:91 course2:84.5 course3:72 course4:88 course5:86.67

```
fileRDD = sc.textFile(inputFile)

courseRDD = fileRDD.map(lambda line: line.split(':')[1]).flatMap(lambda x: x.split(';')).map(lambda x: (x.split(',')[0], x.split(',')[1]))

resRDD=courseRDD.mapValues(lambda  x:(int(x),1)).reduceByKey(lambda a, b: (a[0]+b[0], a[1]+b[1])).mapValues(lambda x:x[0]/x[1]).sortByKey()
```

(ii) (5 marks) **DataFrame programming (RDD APIs not allowed):** Solve the above problem using the DataFrame APIs.

```
fileDF = spark.read.text(inputFile)

student=fileDF.select(split(fileDF['value'],':').getItem(0).alias('sid'),
split(fileDF['value'], ':').getItem(1).alias('courses'))

scDF = student.withColumn('course', explode(split('courses', ';')))

scDF2=scDF.select(split(scDF['course'],',').getItem(0).alias('cname'),
split(scDF['course'], ',').getItem(1).alias('mark'))

avgDF = scDF2.groupBy('cname').agg(avg('mark')).orderBy('cname')
```

Question 4. Mining Data Streams (8 marks)

Consider a Bloom filter of size m = 7 (i.e., 7 bits) and 2 hash functions that both take a string (lowercase) as input:

- h1(str) = ∑(c in str)(c-'a') mod 7
- h2(str) = str.length mod 7

Here, c - 'a' is used to compute the position of the letter c in the 26 alphabetical letters, e.g., h1("bd") = (1 + 3) mod 7 = 4.

(i) Given a set of string S = {"hi", "big", "data"}, show the update of the Bloom filter
(ii) Given a string "spark", use the Bloom filter to check whether it is contained in S.
(iii) Given S in (i) and the Bloom filter with 7 bits, what is the percentage of the false positive probability (a correct expression is sufficient: you need not give the actual number)?

(i)

	hi	big	data
h1	(7+8) mod 7 = 1	(1+8+6) mod 7 = 1	(3+0+19+0) mod 7 = 1
h2	2 mod 7 = 2	3 mod 7 = 3	4 mod 7 = 4

(ii)

h1 (spark) = (18 + 15 + 0 + 17 + 10) mod 7 = 4
h2 (spark) = 5 mod 7 = 5
Not in S since the 4th bit is 1 but the 5th bit is 0

(iii) k – # of hash functions; m – # of inserting elements; n - # of bits

$$(1 - e^{-\frac{km}{n}})^k = 0.3313$$

Question 4. Mining Data Streams (8 marks, another example)

For the heavy hitter problem, given a sequence [1,1,2,3,4,5,1,1,1,5,3,3,1,1,2] and ε=0.2, please use the Lossy Counting algorithm to find out the approximate results of frequent elements in the sequence.

Divide the sequence to three windows [1,1,2,3,4], [5,1,1,1,5], [3,3,1,1,2]

For the first window, we have [1:2, 2:1, 3:1, 4:1]. Decrement all counters by 1 and drop elements with counter 0, and we get [1:1]

For the second window, we have [1:4, 5:2]. Decrement all counters by 1, and we get [1:3, 5:1].

For the third window, we have [1:5, 2:1, 3:2, 5:1]. Decrement all counters by 1, and we get [1:4, 3:1].

Question 5. Finding Similar Items (8 marks)

(i) (2 marks) Given three documents D1 : “abcbacb”, D2 : “cbaaabc”, and D3 : “acbaaac”, using the characters as tokens to compute k-shingles, What is the smallest value of k such that the total number of distinct k-shingles is 9? List all the k-shingles generated from the three documents.

Answer: k=3
D1 has abc, bcb, cba, bac, acb
D2 has ~~cba~~, baa, aaa, aab, ~~abc~~
D3 has ~~acb, cba, baa, aaa,~~ aac

(ii) (5 marks) We want to compute min-hash signature for two columns, C1 and C2 using two pseudo-random permutations of columns using the following function:

- h1(n) = 3n + 2 mod 7
- h2(n) = 2n - 1 mod 7

Row	C_1	C_2
0	0	1
1	1	0
2	0	1
3	0	0
4	1	1
5	1	1
6	1	0

	C1	C2
h1	∞	∞
h2	∞	∞

Scanning Row 0:

	C1	C2
h1	∞	2
h2	∞	6

Scanning Row 1:

	C1	C2
h1	5	2
h2	1	6

Scanning Row 2:

	C1	C2
h1	5	1

h2	1	3

Scanning Row 4:

	C1	C2
h1	0	0
h2	0	0

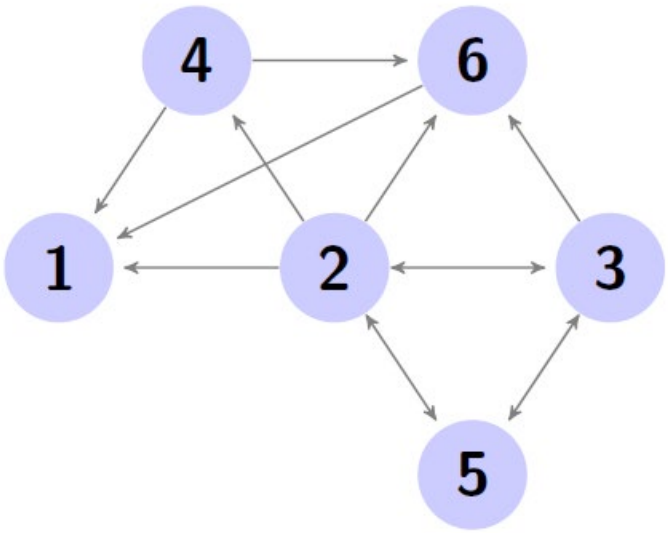
Since all entries are of value 0, no smaller row index can be obtained, and we can stop now.

(iii) (1 mark) Suppose we wish to find similar sets, and we do so by minhashing the sets 10 times and then applying locality-sensitive hashing using 5 bands of 2 rows (minhash values) each. If two sets had Jaccard similarity 0.6, what is the probability that they will be identified in the locality-sensitive hashing as candidates (i.e. they hash at least once to the same bucket)? You may assume that there are no coincidences, where two unequal values hash to the same bucket. A correct expression is sufficient: you need not give the actual number.

$1 - (1 - 0.6^2)^5$

Question 6. Graph Data Management (8 marks)

A directed graph G has the set of nodes {1,2,3,4,5,6} with the edges arranged as follows.



Set up the PageRank equations, assuming β = 0.8 (jump probability = 1- β). Denote the PageRank of node a by r(a).

$$r(1) = 0.8(\frac{1}{6} \cdot r(1) + \frac{1}{2} \cdot r(4) + r(6) + \frac{1}{5} \cdot r(2)) + \frac{0.2}{6} \quad (1)$$

$$r(2) = 0.8(\frac{1}{6} \cdot r(1) + \frac{1}{3} \cdot r(3) + \frac{1}{2} \cdot r(5)) + \frac{0.2}{6} \quad (2)$$

$$r(3) = 0.8(\frac{1}{6} \cdot r(1) + \frac{1}{5} \cdot r(2) + \frac{1}{2} \cdot r(5)) + \frac{0.2}{6} \quad (3)$$

$$r(4) = 0.8(\frac{1}{6} \cdot r(1) + \frac{1}{5} \cdot r(2)) + \frac{0.2}{6} \quad (4)$$

$$r(5) = 0.8(\frac{1}{6} \cdot r(1) + \frac{1}{5} \cdot r(2) + \frac{1}{3} \cdot r(3)) + \frac{0.2}{6} \quad (5)$$

$$r(6) = 0.8(\frac{1}{6} \cdot r(1) + \frac{1}{5} \cdot r(2) + \frac{1}{3} \cdot r(3) + \frac{1}{2} \cdot r(4)) + \frac{0.2}{6} \quad (6)$$

We first deal with the deadend. With probability of 0.8, we follow the graph structure. With probability 0.2, we perform the teleport.

End of Examination Paper