COMP9313 2022T3 Exam

The deadline for the final exam is:

Monday 28th, November 5:00 pm (AEDT)

Please submit your answers through Moodle. Do not wait till the last minute to submit and double check if it is successful. Please send emails to xin.cao@unsw.edu.au and siqing.li@unsw.edu.au if you have any questions.

Question 1. Concepts (4 marks)

(a) (2 marks) In what kind of problems a combiner class and a reducer class can be used interchangeably? Please use an example to explain your answer.

(b) (2 marks) In one project, a student complained that her approach took a lot of time at the step when using the reduce() function, but all the previous operations including reading the data by textFile(), filtering the data by filter(), and transform the data by map() and flatmap(). Could you please explain the reason to her?

Question 2. MapReduce Programming (14 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 the classes including Mapper and Reducer (optionally Combiner etc.if necessary, and the* ***efficiency*** *of your method will be considered).*

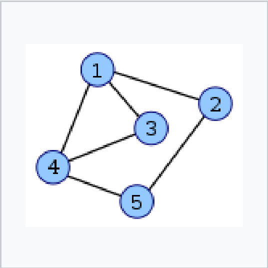
(a) (4 marks) Given a table shown as below, find out the person(s) with the maximum salary in each department (employees could have the same salary).

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

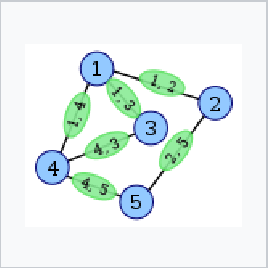
(b) (10 marks) **Problem Background:** Given an undirected graph G, its “line graph” is another graph L(G) that represents the adjacencies between edges of G, such that:

- each vertex of L(G) represents an edge of G; and
- two vertices of L(G) are adjacent if and only if their corresponding edges share a common endpoint ("are incident") in G.

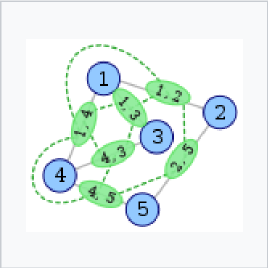
The following figures show a graph (left) and its line graph (right). Each vertex of the line graph is shown labelled with the pair of endpoints of the corresponding edge in the original graph. For instance, the vertex on the right labelled (1,3) corresponds to the edge on the left between the vertices 1 and 3. Vertex (1,3) is adjacent to three other vertices: (1,2) and (1,4) (corresponding to edges sharing the endpoint 1 in G) and (3,4) (corresponding to an edge sharing the endpoint 3 in G). Note that the vertex (4, 3) in the below example should be (3, 4) in the output.



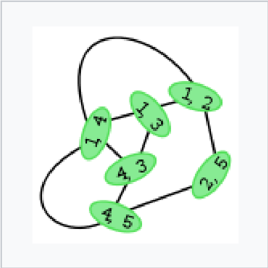
Graph G



Vertices in L(G) constructed from edges in G



Added edges in L(G)



The line graph L(G)

Problem: Given you the adjacency list of an undirected graph G, use MapReduce to generate the adjacency list of its line graph L(G). Note that each edge connecting two nodes i and j is represented by (i, j) in L(G) (if i<j). In the output, the edges in each list should be ranked in ascending order by comparing the first node and then the second node. The adjacency lists should be ranked by the keys according to the same order as well. Take the above figure as an example, sample input and output are as below:

Input:	Output:
1: 2, 3, 4	(1, 2): (1, 3), (1, 4), (2, 5)
2: 1, 5	(1, 3): (1, 2), (1, 4), (3, 4)
3: 1, 4	(1, 4): (1, 2), (1, 3), (3, 4), (4, 5)
4: 1, 3, 5	(2, 5): (1, 2), (4, 5)
5: 2, 4	(3, 4): (1, 3), (1, 4), (4, 5)
	(4, 5): (1, 4), (2, 5), (3, 4)

Question 3. Spark Programming (14 marks)

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

(a) (7 marks) **RDD programming:** Given a set of marks from different courses (the input format is as shown in the left column), the task is to: For each student, get his/her ranking in different courses. The output format is <student_id: course_name, rank>. Sort the output by student_id first and then by course_name (the format is as shown in the right column).

Input:	Output:
student1 course1 90 student1 course2 92 student1 course3 80 student1 course4 79 student1 course5 93 student2 course1 92 student2 course2 77 student2 course5 85 student3 course3 64 student3 course4 97 student3 course5 82	student1: course1,2 student1: course2,1 student1: course3,2 student1: course4,2 student1: course5,1 student2: course1,1 student2: course2,2 student2: course5,2 student3: course3,2 student3: course4,1 student3: course5,3

(b) (7 marks) **DataFrame programming (RDD APIs not allowed):** Given the same input (but different format!) as in problem (a), 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

Question 4. Finding Similar Items (6 marks)

(a) (2 marks) Given two documents A = (“the sky is dark the moon is bright”) and B = (“the moon in the sky is bright”), using the ***words*** as tokens, compute the 2-shingles for A and B, and then compute their Jaccard similarity based on their 2-shingles.

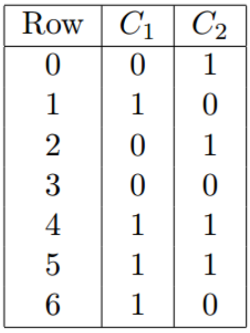
(b) (3 marks) We want to compute min-hash signature for two columns, C1 and C2 using two pseudo-random permutations of columns using the following function:

h1(n) = (5n + 2) mod 7

h2(n) = (3n + 1) mod 7

Here, n is the row number in original ordering. Instead of explicitly reordering the columns for each hash function, we use the implementation discussed in class, in which we read each data in a column once in a sequential order, and update the min hash signatures as we pass through them.

Complete the steps of the algorithm and give the resulting signatures for C1 and C2.



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

(c) (1 marks) 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.

Question 5. Mining Data Streams (6 marks)

(a) (3 marks) Counting Bloom Filter

Consider a Counting Bloom filter of size m = 7 and 2 hash functions that both take a string (lowercase) as input:

h1(str) = $\sum_{c\ in\ str}(c - 'a')$ mod 7

h2(str) = (str.length * 2) 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) (2 marks) Given a set of string S = {“hi”, “big”, “data”, “spark”}, show the update of the Bloom filter

(ii) (1 mark) Delete “hi” from S, and then use the bloom filter to check if “sql” is contained in S.

(b) (3 marks) CM-Sketch

Assume that we have 5 buckets and three hash functions:

h0(str) = (str.length * 2) mod 5

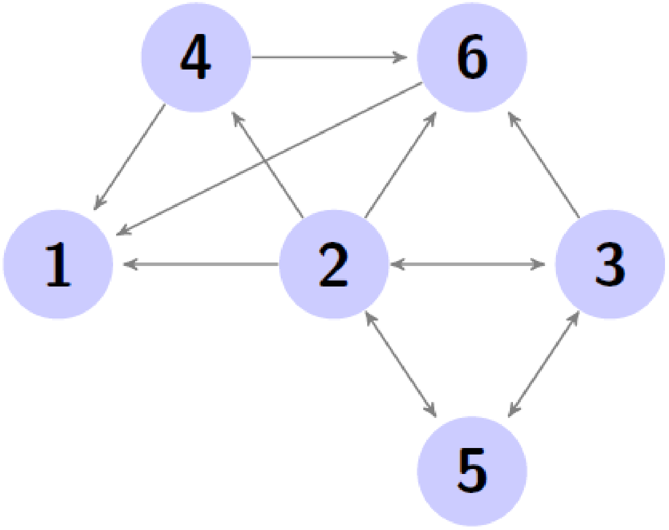
h1(str) = (str.length + 3) mod 5

h2(str) = (str[0]-‘a’) mod 5

Given you a stream of terms: “big”, “data”, “data”, “hadoop”, “data”, “spark”, show the steps of building the CM-Sketch. Then, use the built CM-sketch to get the count for word “data”.

Question 6. Link Analysis (6 marks)

Given a directed graph G with the set of nodes {1,2,3,4,5,6} and the edges arranged as below:



Using the MapReduce PageRank algorithm (lecture slides 9.50 and 9.51), show the computation process in the first two rounds (including the mapper input, mapper output, reducer input, and reducer output).

End Of Paper