CSE6332 Cloud Computing& Big Data Fall 2022

Final Exam

December 13, 2022

Answer all questions in the space provided next to the question. Keep your answers brief and precise. It is open notes but closed books. You are allowed to bring up to 40 pages of notes stapled together in **one** folder. You are not allowed to remove any of the pages from your notes or the exam paper during the exam. You are allowed to use pens, pencils, and erasers only. You may use the back pages of the exam for scratch paper. 100 points possible. This exam is worth 32% of your final grade. Duration: 2 hours. Good luck!

FIRST-NAME LAST-NAME:

STUDENT ID:

1. *(20 points)* Assume that you have two files `students.txt` and `enrollment.txt`. The `students.txt` file is a CSV text file, where each line represents a student with a student ID, a name, and an address. You may assume that both the student ID and the name are unique. The `enrollment.txt` is a CSV text file that contains the course enrollment, where each line has a course code, a student ID, and the score (0-100) that this student received in this course. For example:

students.txt:

```
100035,John Smith,102 Main St
100044,Mary Jones,45 Division St
```

enrollment.txt:

```
cse6332,100035,85
cse6332,100044,100
cse5335,100035,98
```

Write Map-Reduce pseudo code to answer the following query: for each different student, print the student name and the average score of this student on all his/her classes. You only need to write the mappers and reducers of your Map-Reduce jobs using pseudo-code (not Java). You don't need to write the main program or the run method.

2. *(20 points)* Write Spark code in Scala to read and parse the data from the text files `students.txt` and `enrollment.txt`. Then write Spark code in Scala to implement Problem 1: for each different student, print the student name and the average score of this student on all his/her classes. The code must print the results to the output. You don't need to write the complete code (that is, don't write code to create the Spark context).

3. *(15 points)* Write Pig code to read and parse the data from the text files `students.txt` and `enrollment.txt`. Then write Pig code to implement Problem 1: for each different student, print the student name and the average score of this student on all his/her classes. You may store the results in a file.

4. *(20 points)* Write Spark code in Scala that creates two DataFrames to store the data from `students.txt` and `enrollment.txt`. Then, write a Spark SQL query to implement Problem 1: for each different student, print the student name and the average score of this student on all his/her classes. The query must print the results to the output. You don't need to write the complete code (that is, don't write code to create the Spark context).

5. *(25 points total)* A directed graph is represented in an input text file `graph.txt` using one line per graph vertex. For example, the line

```
1,2,3,4,5,6,7
```

represents the vertex with ID 1 that has outgoing links to the vertices with IDs 2, 3, 4, 5, 6, and 7.

(a) *(5 points)* Write Spark code in Scala that reads the file `graph.txt` and returns an RDD of type `RDD[(Int,Int)]` that contains a pair (i, j) for each link from i to j.

(b) *(20 points)* Write Spark code in Scala that finds all the loops of size 3 in the graph. That is, it finds all triples of vertices i, j, and k (they don't have to be different vertices) such that there is a link from i to j, a link from j to k, and a link from k to i. Your Spark program should print all these triples (i, j, k) to the output.

You don't need to write the complete code (that is, don't write code to create the Spark context).