Quiz 1: Introduction/MapReduce Name: ______________________ ID: ________________

Q1. Write a MapReduce program that counts the number of integers in a given group of integers. For example, if the input is {1, 3, 2, 5, 3, 4}, the program should output 6. <u>You can but not required to use a Combiner in this program.</u> Pseudo code is fine, but make sure you indicate the Map and Reduce functions, and Combiner (if any) in the program, and the input, output, and logic of each component. (3 pts)

```
1) map(key, value)
    # value is the group of integers
      for v in value:
          emit (1, v)

2) Reducer (key, value)
    # key : 1 , value : list of integers
    sum = 0
    for v in value
        sum += 1
    emit ( 1, sum)
```

RUBRIC :

1) 1 mark for input & output
2) 1 mark for the logic
3) 1 mark for pseudocode.

Q2. Briefly explain how we could approximate $(1+a)^b$ as e^{ab} (2 pts) (chapter 1.3.5)

Assume a is small

$(1+a)^b$ can be written as $(1+a)^{(1/a)(ab)}$

substitute $a = 1/x$

$\therefore (1+\frac{1}{x})^{xab}$; $(1+\frac{1}{x})^x$ is close to the value of e

thus, $(1+a)^b \approx e^{ab}$.

RUBRIC

1) 0.5 marks if its close.
2) 2 marks if everything is right.

Q3. What is power law? Give an example (1 pt) (chapter 1.3.6)

Power law is the functional relationship between 2 quantities. One quantity varies over the power of other. ⇒ $\boxed{y = cx^a}$

example: Book sales at Amazon.com, where x represents rank of book sales & y is the no. of sales of the x^{th} best-selling book over some period.

RUBRIC

0.5 for power law

0.5 for example

Q4. Briefly explain the benefit of a distributed file system (2 pts)

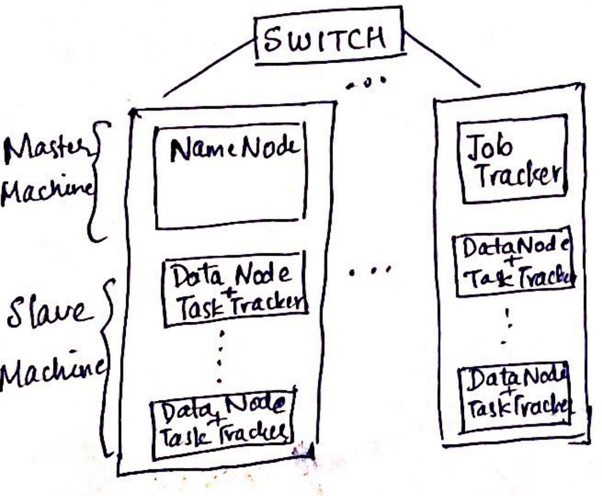
RUBRIC

→ High availability - Stores data redundantly on multiple nodes for persistence and availability

→ Moves computation close-to data to minimize data movement.

→ 1 mark each for 2 benefits

Q5. Draw a simple diagram of connected nodes (machines) and their functionality (e.g., master nodes and slave nodes as in the slides) in a map reduce environment (2 pts)



NameNode : Oversees the health of datanode & coordinates the access to data

Job tracker : Coordinate the parallel programming of data using map reduce.

slave nodes : responsibles for storing the data and processing the computation

Task Tracker : manages the processing resources on each slave node

RUBRIC

→ 1 mark for diagram

→ 1 mark for functionality