Name (Last, First): _____

This exam consists of 5 questions on 7 pages; be sure you have the entire exam before starting. The point value of each question is indicated at its beginning; the entire exam is worth 100 points. Individual parts of a multi-part question are generally assigned approximately the same point value: exceptions are noted. This exam is open text and notes. However, you may NOT share material with another student during the exam.

Be concise and clearly indicate your answer. Presentation and simplicity of your answers may affect your grade. **If I cannot read your answer easily you will not get credit**. Answer each question in the space following the question. If you find it necessary to continue an answer elsewhere, clearly indicate the location of its continuation and label its continuation with the question number and subpart if appropriate.

You should read through all the questions first, then pace yourself.

The questions begin on the next page.

Problem	Possible	Score
1	20	
2	20	
3	20	
4	20	
5	20	
Total	100	

1. (_____/20 points)

   Distributed Sorting Revisited

   Consider Project 1. Now assume that the 10GB file exists on both machines rather than on the first machine. However, you still need to end up with a 10GB sorted file on the first machine. How would you design Project 1 given this change?

2. (_____/20 points)

Leader Election

Recall the paper *Elections in a Distributed Computing System* by H. Garcia-Molina.

(a) (10 points) In the Bully Algorithm, once a a process becomes a leader, will it remain as the leader until it stops or fails? Explain your answer and provide details from the paper to support your answer.

(b) (10 point) In the Invitation Algorithm, is it possible to have multiple leaders in the event there is a network partition? For example, assume there are five processes: A, B, C, D, and E. Initially all the processes are connected, but later a network partition occurs and only A and B can communicate with each other and not with C, D, and E. Processes C, D, and E can all communicate with each other. Will two leaders be elected in this case? Explain your answer and provide details from the paper to support your answer.

3. (_____/20 points)

ZooKeeper

Recall the paper *ZooKeeper: Wait-free coordination for Internet-scale systems* by P. Hunt, M. Konar, F. P. Junqueira and B. Reed. In Section 2.4, the authors show how to implement Read/Write Locks using the ZooKeeper primitives:

```
Write Lock
1 n = create(l + /write-, EPHEMERAL|SEQUENTIAL)
2 C = getChildren(l, false)
3 if n is lowest znode in C, exit
4 p = znode in C ordered just before n
5 if exists(p, true) wait for event
6 goto 2

Read Lock
1 n = create(l + /read-, EPHEMERAL|SEQUENTIAL)
2 C = getChildren(l, false)
3 if no write znodes lower than n in C, exit
4 p = write znode in C ordered just before n
5 if exists(p, true) wait for event
6 goto 3
```

In class we determined that this implementation gives preference to writers. That is, as soon as a writer tries to acquire the lock, no further reader will be allowed to acquire the lock.

Show how to modify this Read/Write lock implementation so that readers can always acquire the lock as lock as other readers currently hold the lock. In this way, they only way a writer can acquire the lock is if there are no readers. Readers that come after a writer acquires the lock will have to wait for the writer to release the lock.

4. (_____/20 points)

Raft

Recall the paper *In Search of an Understandable Consensus Algorithm* by D. Ongaro and J. Ousterhout.

(a) (5 points) Consider the following log on a Raft server.

```
log index  1   2   3   4   5   6
term     | 1 | 1 | 3 | 2 | 2 |
```

Is this log configuration possible? In the answer is "no", explain why not.

(b) (5 points) Consider the following log on a Raft server. Only the term numbers are given, not the values.

```
log index  1   2   3   4   5   6   7
term     | 1 | 1 | 2 | 2 | 2 | 3 |
```

Is this log configuration possible? In the answer is "no", explain why not.

(c) (10 points) In Raft, log compaction is achieved through snapshotting. Describe what you think happens if a server fails while taking a snapshot? That is, how can we know the snapshot is valid and that we can remove old log entries?

5. (_____/20 points)

The Google File System

Recall the paper *The Google File System* by Ghemawat, Gobioff, and Leung.

One issue with GFS and also HDFS is that the master is a single point of failure. In GFS, a log with commits to secondaries are used to ensure a fast restart in the presence of failure of the master. However, it might be better to support continuous operation in the presence of a master failure.

Propose and justify a design for GFS utilizing ZooKeeper to support continuation operation of a master server. That is we don't want require a master restart.

Continue your answers here if necessary.