This exam is open book and notes. You can use a calculator and your laptop to access course notes and videos (but not to communicate with other people). You have 130 minutes (2 hours, 10 minutes) to complete it.

Print your name:______________________________

The Honor Code is an undertaking of the students, individually and collectively:

1. that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

2. that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

I acknowledge and accept the Honor Code.

Signed:____________________________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td></td>
</tr>
</tbody>
</table>
Problem 1 (10 points)

State if the following statements are TRUE or FALSE.

(a) In Chord, if there are $K$ keys and $N$ nodes, each node will be assigned exactly $N/K$ keys.
   True or false?:

(b) When Zookeeper is used, it is possible for a client to connect to any server and read any object stored in the system.
   True or false?:

(c) Consider an inverted distributed index (for document retrieval) that is split across documents. (That is, each partition indexes a disjoint subset of the documents.) In this case, a query must access all partitions, but adding one document only involves a single partition.
   True or false?:

(d) Consider a distributed publish/subscribe system that uses $N$ computers. It is always necessary to have each subscription stored at all $N$ computers, so it can be matched against new publications.
   True or false?:

(e) Hive is an efficient implementation of Map-Reduce on top of a relational database.
   True or false?:

(f) Hive is scalable as long as the queries refer to a few rows from each table, and do not require full-table scanning.
   True or false?:

(g) Spanner is more scalable than BigTable because it is based on a weaker notion of consistency than BigTable.
   True or false?:

(h) Spanner can infer the order of two events happening in two different data centers by taking into account the events’ timestamps.
   True or false?:

(i) S4, Map-Reduce and Hyracks all use the dataflow computation model.
   True or false?:

(j) Dryad is capable of executing MapReduce jobs.
   True or false?:
Problem 2 (10 points)

A data item $X$ is replicated at 4 nodes: $a, b, c, d$. Let us assume these nodes are connected over a reliable network. These nodes, however, may fail. Let the probability that node $i$ is operational at a given time be $P(i)$. These probabilities are independent of each other. In particular, let us assume that

- $P(a) = 0.5$;
- $P(b) = 0.8$;
- $P(c) = 0.9$;
- $P(d) = 1.0$;

A three-phase commit protocol is used for write transactions to $X$, and we use a coterie to define the subsets of nodes that are able to terminate transactions.

(a) Assume we use the following coterie:

$C = \{\{a, b\}, \{b, c, d\}, \{a, c, d\}\}$.

Calculate the probability that a write transaction can complete successfully. (Assume that during the time it takes to execute the transaction, a node $i$ is fully operational with probability $P(i)$, else it is unavailable for the full period with probability $1 - P(i)$.)

Probability:______________

(b) Does the write coterie in part (a) have a vote assignment? If yes, please specify. If no, explain why.

Does coterie have votes assignment? (Yes/No):______________

Assignment or explanation:

(c) Determine a write coterie that maximizes the probability that a write transaction is successful (under the $P(a)$, $P(b)$, $P(c)$, $P(d)$ values given above). Please specify the write coterie and the probability that a transaction is successful.

Write coterie:______________

Probability:______________
Problem 3 (10 points)

This problem deals with the allocation of graph nodes to physical resources in a system like Pregel. In particular, consider two machines $A$ and $B$. Our objective is to allocate the nodes of an input graph to $A$ or $B$, so that the overall cost is minimized. For this problem the cost is given by

- Cost = communicationCost + imBalanceCost;
- communicationCost = the number of graph edges that go across machines;
- imBalanceCost = $10 \times \frac{\text{max}}{\text{min}}$;

where $\text{max}$ is the maximum number of graph nodes allocated to a machine, and $\text{min}$ is the minimum number of nodes allocated to a machine. For example, in the graph below, say we allocate nodes $\{a_1, a_2, a_3, a_4\}$ to $A$ and $\{b_1, b_2, b_3, b_4, b_5\}$ to $B$. Then the communication cost is 2, and the imBalance cost is $10 \times \frac{5}{4} = 12.5$. 

![Graph Diagram](attachment:graph.png)
(a) Give the allocation of nodes into $A$ and $B$, which minimizes the overall cost, for this graph:

Write your answer on the copy of the graph below by circling the nodes that are allocated to machine $A$. For instance, if you draw two circles, one around $a_8$, $a_9$, $a_{10}$ and another around $c_8$, $c_9$, $c_{10}$ then you are indicating that the nodes inside the circles, i.e., $a_8$, $a_9$, $a_{10}$, $c_8$, $c_9$, $c_{10}$, go on machine $A$, and the rest go on machine $B$. Also, give the communication and imBalance costs for your selected configuration.

Note: Please do not write anything else on the graph below, just the circles that constitute your answer. Use the previous copy of the graph to scribble on and work out your solution.

communicationCost: _______________ ; imBalanceCost: _______________
(b) Next consider a complete graph with 10 nodes (a 10-clique). (Every node has a direct connection to all other nodes.) To minimize the total cost, how many nodes should we allocate to machine $A$ and how many to machine $B$?

nodes on machine $A$: ___________; nodes on machine $B$: ___________
Problem 4 (10 points)

(a) Consider the following Pig Latin program:

\[ A2 = \text{FILTER} \ A1 \ \text{BY} \ Y > 10 \]

\[ G1 = \text{COGROUP} \ A2 \ \text{BY} \ X, \ B1 \ \text{BY} \ Y \]

and the following two input tables.

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
</tr>
</tbody>
</table>

Table A1:  
Table B1:  

In the table below, please write the contents of G1 for the above input data. Be sure to write the column names in the first row (above the double lines). Note that the table may have more rows that what you need.

(b) We want to find values that occur the same number of times in A2.X as in B1.Y. For our sample tables from Part (a), the value “7” is one of the values we want because it occurs 2 times in A2.X and two times in B1.Y. Write a single Pig statement that finds these values for any A2, B1 tables. The only input for this statement should be G1 as computed in Part (a). For each of the desired values there should be one tuple in the output, where the first value in the tuple is that value. (The tuple may contain other attributes/columns.) Your statement can use the SUM function (explained in the Pig paper and in class notes.)

Answer (single Pig statement):
Problem 5 (10 points)

You are building a peer-to-peer system. The system has a fixed number $C$ of clients (regular peers), and will have $S$ super-peers. Each super-peer will have a direct connection to all other super-peers (super-peers form a fully connected graph), and will service $C/S$ clients. Your goal is to determine a good number for $S$.

There are three conflicting factors you need to consider.

- **Network Cost:** The first cost is for connecting the super-peers. We model this factor as $S(S-1)/2$, the number of connections required.

- **Super-Peer Cost:** This is the cost of the super-peers themselves, proportional to $S$.

- **Delay Cost:** As we have fewer super-peers, each super-peer services more clients and provides worse service, i.e., longer query delays for the clients. We represent this delay factor by $C/S$, i.e., we expect delays to decrease linearly as we increase the number of super-peers.

To compute the overall cost we need to specify how important each type of cost is, relative to the other costs. Let us say that network and super-peer costs are equivalent, and that delay costs are $k$ times more critical than the other two costs, so our total cost formula becomes

$$T = \frac{S(S-1)}{2} + S + k(C/S).$$

Our goal is to minimize this total cost.

(a) What is the total cost when we have 100 regular peers, 5 super-peers and $k = 1$?

   Total Cost: ____________

(b) How many super-peers should we allocate to minimize the total cost, when there are 100 regular peers and $k = 1$?

   Optimal $S$: ____________

(c) How many super-peers should we allocate to minimize the total cost, when there are 100 regular peers and $k = 2$?

   Optimal $S$: ____________
Problem 6 (10 points)

There are two parts to this timestamp-ordering problem: part (a) deals with a 2-version mechanism, while part (b) compares strict versus not-strict mechanisms.

(a) In a 2-version timestamp ordering concurrency control mechanism the system maintains two version for each data object $X$: the current version $X$, and one previous version, which we call $X^{-1}$.

When a transaction issues a write request $w_i(X)$, the system checks $T_i$’s timestamp against the timestamp of the last read and write on $X$ as usual. If the check is successful, then the write $w_i(X)$ is executed: first the system stores the value of $X$ into $X^{-1}$ and then it writes the new value into $X$. Note that at system startup, both $X$ and $X^{-1}$ hold the initial value of $X$.

When a transaction issues a read request $r_i(X)$, the system can either read $X$ or $X^{-1}$, as long as the timestamp ordering rule is not violated.

Consider a scenario where 5 transactions $T_1, T_2, T_3, T_4, T_5$, access an object $X$, and the timestamps are ordered this way:

$$ts(T_1) < ts(T_2) < ts(T_3) < ts(T_4) < ts(T_5).$$

Consider the schedules in the following table. Using the rightmost two columns, indicate if a schedule is feasible (write “Y”) or not feasible (write “N”) for a 1-version system (column 2), or a 2-version system (column 3). (A 1-version system is what we covered in class.)

<table>
<thead>
<tr>
<th>Schedule</th>
<th>1-Version</th>
<th>2-Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1 : w_1(X)w_3(X)r_4(X)r_2(X)w_5(X)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_2 : w_1(X)w_3(X)w_5(X)r_4(X)r_2(X)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_3 : w_1(X)r_4(X)r_2(X)w_3(X)w_5(X)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_4 : w_1(X)r_2(X)w_3(X)r_4(X)w_5(X)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_5 : w_1(X)w_3(X)r_2(X)w_5(X)r_4(X)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_6 : w_1(X)r_2(X)w_5(X)w_3(X)r_4(X)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
(b) In class we initially discussed a simple timestamp ordering scheme, and then introduced a
variant called strict TO. To compare the two schemes, consider two transactions $T_2$ and $T_3$
with timestamps ordered this way: $ts(T_2) < ts(T_3)$.

Consider the schedules in the following table. In these schedules, we assume that if an action
$p_i()$ occurs in the schedule before another action $q_j()$, the $p_i()$ action occurred before $q_j()$
did (even if the actions execute at different nodes). (Note that transactions $T_2$ and $T_3$ may
consist of different actions in each scenario.)

In the same table, indicate if a schedule is feasible (write “Y”) or not feasible (write “N”)
for a simple TO scheduler (column 2) and for a strict TO scheduler (column 3).

<table>
<thead>
<tr>
<th>Schedule</th>
<th>Simple TO</th>
<th>Strict TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{11} : w_2(X)r_3(X)w_2(Y)w_2(Z)r_3(Z)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{12} : w_2(X)w_2(Y)w_2(Z)r_3(X)r_3(Z)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{13} : w_2(X)w_3(X)w_2(Y)w_2(Z)r_3(Z)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$S_{14} : r_3(X)r_3(Z)w_2(X)w_2(Y)w_2(Z)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem 7 (10 points)

PhotoNet, a social networking service, carries a fabulous photo gallery posted by its registered users. A registered user is identified by his/her unique user ID. We assume a registered user has exactly one user ID, in other words, we ignore the case that a user has more than one user ID. A user can post his/her photos and/or hit “like” buttons on others’ photos. We also assume that, no matter how much a user likes a photo, he/she “likes” each photo at most once.

The underlying database has three tables:

- user(user.ID, contact.info); key is user.ID
- photo(photo.ID, author.ID, description, link_to_photo); key is photo.ID
- like(photo.ID, fan.ID, like.date); key is the pair photo.ID, fan.ID

Note that author.ID contains the user.ID of the user who took the photo, while fan.ID is the user.ID of the user who likes the photo.

PhotoNet is hosting a contest in which the photo with the highest number of “likes” will win a big prize. The contest is so popular that all tables become huge, so you are asked to design a Map reduce job to find the winner.

(a) Your first task is to count the number of likes each photo gets. The output should be a file (possibly partitioned across several sites); for each photo.ID P, the file should contain a tuple [P, N] where N is the total number of “likes” of P. If photo P had NO likes, there should be NO [P, 0] tuple in the output.

Write the map and reduce functions using pseudo-code; clarity is important. Also indicate the input table(s) used. Use dot notation to refer to fields of tuples. For instance, if your map function reads a tuple t1 from table R(A,B), then t1.A is the value of the A attribute, and t1.B is the value of the B attribute. If necessary, you can use t1.rel to refer to the name of the table that t1 came from (i.e., R).

Input table(s): ______________

Map function, map(t):

Reduce function, reduce(key, val):

(b) Next we want to generate a sorted file of the total likes counts. For example, if the output of part (a) was

\[[P1, 5], [P2, 3], [P3, 5], [P4, 1], [P5, 1], [P6, 9]\]

then the result should be

\[[9, \{P6\}], [5, \{P1, P3\}], [3, \{P2\}], [1, \{P4, P5\}]\].

The result must be sorted by total like counts, either in ascending or descending order. The result could be split across sites; for example we could have \[[9, \{P6\}], [3, \{P2\}]\] on one site and \[[5, \{P1, P3\}], [1, \{P4, P5\}]\] on another. In any of these cases, it is easy to find the photo(s) with the maximum number of likes.

Write below the map and reduce functions to generate this sorted file.

Input table(s): the output of Part (a) (call attributes \texttt{photo-ID} and \texttt{count})

Map function, \texttt{map(t)}:

Reduce function, \texttt{reduce(key, val)}:
(c) PhotoNet would also like to reward the users who like the photo with the highest number of likes but do NOT like any other photos. (That is, a winner only likes a single photo, i.e., the photo with the most likes.) Your last task is to list the user IDs of all the users to be rewarded. The output should be a set of tuples of the form [user_ID], where user_ID is one of the winning user(s).

Assume that there is a single photo with the highest number of likes, and that photo has photo_ID = best. Value best was found in Part (b) and can be used in your map, reduce code below.

Input table(s): 

Map function, map(t):

Reduce function, reduce(key, val):