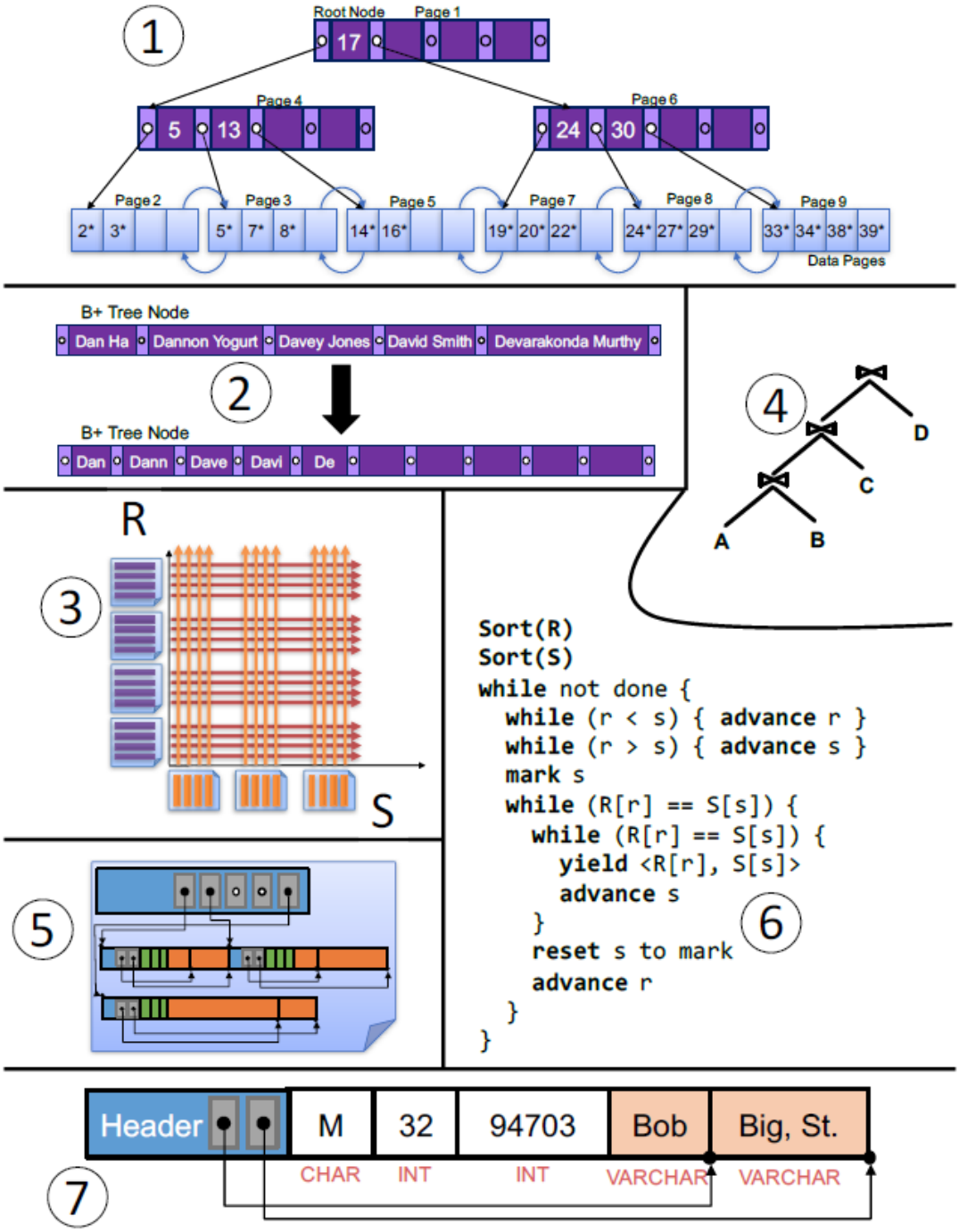
CS150: Database and Data Mining
Final Exam Solutions

January 8, 2021

I Basics [10 points]

For each image on the next page, select the letter corresponding to the best description.
A. Left Deep Tree
B. Key Compression
C. B+ Tree
D. ISAM
E. Nested Loops Join
F. Sort Merge Join
G. Indexed Nested Loop Join
H. Slotted Page
I. Variable Length Tuple
J. Fixed Length Tuple
K. Buffer Frame
L. Sort based group by
M. mapPartitions
N. Tournament Sort

Solution
1. C (not an ISAM because pages non-sequential and prev/next pointers present)
2. B (only prefixes are needed to search tree)
3. E (a nested loops join goes through every possible pair of tuples)
4. A (in fact, this is a join plan)
5. H (there is a slot header holding pointers to records)
6. F (this requires sorting, and it also yields pairs of records between two relations)
7. I (the pointers at the record header point to the end of variable-length strings)


1
Root Node
Page 1
17
Page 4
5
13
Page 6
24
30
Page 2
2*
3*
Page 3
5*
7*
8*
Page 5
14*
16*
Page 7
19*
20*
22*
Page 8
24*
27*
29*
Page 9
33*
34*
38*
39*
Data Pages
B+ Tree Node
Dan Ha
Dannon Yogurt
Davey Jones
David Smith
Devarakonda Murthy
2
B+ Tree Node
Dan
Dann
Dave
Davi
De
4
D
C
A
B
R
3
S
Sort(R)
Sort(S)
while not done {
while (r < s) { advance r }
while (r > s) { advance s }
mark s
while (R[r] == S[s]) {
while (R[r] == S[s]) {
yield <R[r], S[s]>
advance s
}
reset s to mark
advance r
}
}
5
6
Header
M
32
94703
Bob
Big, St.
CHAR
INT
INT
VARCHAR
VARCHAR
7

II SQL and ER Modeling [12 points]

1. **[2 points]** Which of the following expressions computes the matrix vector product:

$$(\mathbf{Ax})_i = \sum_{k=1}^{d} A_{ik} x_k$$

Assume **A** and **x** have compatible dimensions and there is only one correct answer.
A.

```
SELECT A.row AS row, A.value * x.value AS value
FROM A JOIN x
ON A.row = x.row
```

B.

```
SELECT A.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.row = x.row
GROUP BY A.col
```

C.

```
SELECT x.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.col = x.row
GROUP BY A.col
```

D.

```
SELECT A.row AS row, SUM(A.value * x.value) AS value
FROM A JOIN x
ON A.col = x.row
GROUP BY A.row
```

Solution
D: The final answer should be grouped by the row of A and therefore sum over the product along the columns.

2. **[2 points]** Suppose we wanted to compute the element-wise sum of the vectors **x** and **y** using the SQL expression:

```
SELECT x.row AS row, SUM(x.value + y.value) AS value
FROM x JOIN y
ON x.row = y.row
```

Which of the following statements about this query are true? (You may mark zero (ϕ), one or more than one of the choices.)
A. Some non-zero entries may be omitted from the final result.
B. The correct query should use LEFT OUTER JOIN.
C. The correct query should use FULL OUTER JOIN.
D. There is nothing wrong.

Solution
A, C: There was a typo in this question (sum should be removed). The issue with this query is the combination of two factors. First, the two vectors may have zeros in different places, and in a sparse vector the zero entries are represented by the absence of information (missing row IDs in the table). Second, by default, joins in SQL will discard a tuple from one table unless it has a match in the other table; OUTER JOINs preserve these non-matching tuples.
Consider the following:

Vector a

Row	Value
1	3.0
3	4.0

Vector b

Row	Value
2	2.0
5	8.0

These two tables produce no outputs on a typical (inner) join, representing a vector of all 0's. We need to do a full outer join to ensure that we get rows 1, 2, 3 and 5 in the output.

3. **[8 points]** There are four tables. SALESPERSON contains the names, ids & quotas for the salespeople, and names are not unique. PRODUCTS contains the product names, product ids, and prices for the products. The product ids are unique. CUSTOMERS contains the customer names, customer ids, and regions for the customers (customer ids are unique), and ORDERS contains the customer id, the product id, and the product ordered per customer.

SALESPERSON

Sname	Sid	Quota
Frances	25	$100
Bob	31	$150
Frances	74	$200
Mary	89	$250

PRODUCTS

Pname	Pid	Pprice
disks	131	$100
pcs	152	$700
macs	831	$800
printers	255	$120
paper	221	$5

CUSTOMERS

Cname	Cid	Region
Bob	1	TX
Harry	2	TX
Lin	3	MA
Martha	4	FL
Lin	5	FL
Leyla	6	CA

ORDERS

Cid	Pid	Quantity
1	152	1
2	152	1
4	831	1
4	131	1
5	255	1
6	831	1

(a) **[2 points]** Select the true SQL expression(s) for "List the names of the customers who have bought more than one item." (You may mark zero (ϕ), one or more than one of the choices.)

A.

```
SELECT cname
FROM customers
WHERE cid IN (SELECT cid
              FROM orders
              GROUP BY cid
              HAVING count(*) > 1)
```

B.

```
SELECT c.cname
FROM customers c, (SELECT cid
                   FROM orders
                   GROUP BY cid
                   HAVING count(*) > 1) as o
WHERE c.cid = o.cid
```

C.

```
SELECT cname
FROM customers
WHERE (SELECT count(*)
       FROM orders
       GROUP BY pid) > 1
```

D.

```
SELECT cname
FROM customers c, orders o1, orders o2
WHERE c.cid = o1.cid and c.cid = o2.cid and o1.cid <> o2.pid
```

Solution
A, B: In C, "group by pid" is wrong.

(b) **[2 points]** Select the true SQL expressions for "List the names, pid, and price of all the products, whether or not the product has been ordered, but if it has been ordered by the cids of the customer who ordered it." (You may mark zero (ϕ), one or more than one of the choices.)

A.

```
SELECT name, pid, price, cid
FROM products LEFT OUTER JOIN orders
```

B.

```
SELECT name, pid, price, cid
FROM products LEFT OUTER JOIN orders
ON products.pid = orders.pid
```

C.

```
SELECT name, p.pid, price, cid
FROM products p, orders o
WHERE p.pid = o.pid
UNION
SELECT name, pid, price
FROM products p
WHERE NOT EXISTS (SELECT * FROM orders o WHERE o.pid = p.pid)
```

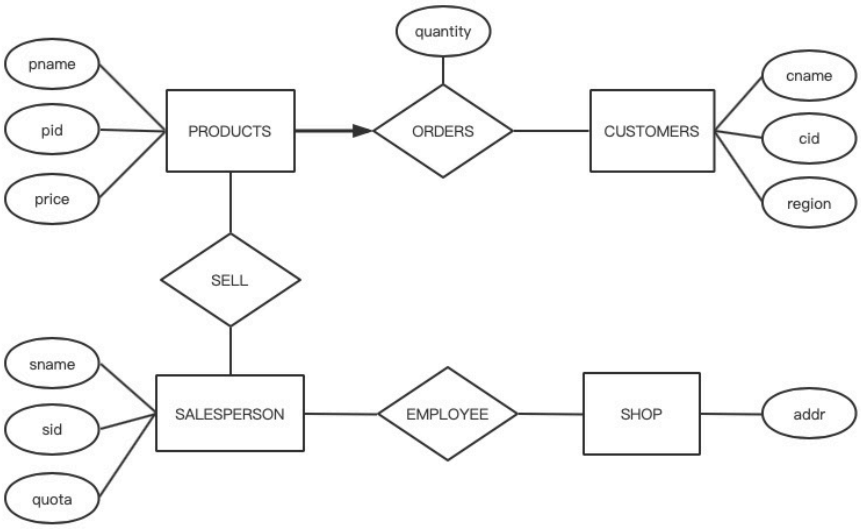
D.

```
SELECT name, p.pid, price, cid
FROM products p, orders o
WHERE p.pid = o.pid
UNION
SELECT name, pid, price, NULL as cid
FROM products p
```

Solution
A, B: In C, should be "SELECT name, pid, price ,NULL as cid"; In D, should add "WHERE NOT EXISTS (SELECT * FROM orders o WHERE o.pid = p.pid)".
We also accept ϕ, because none of the four tables has the field "name".

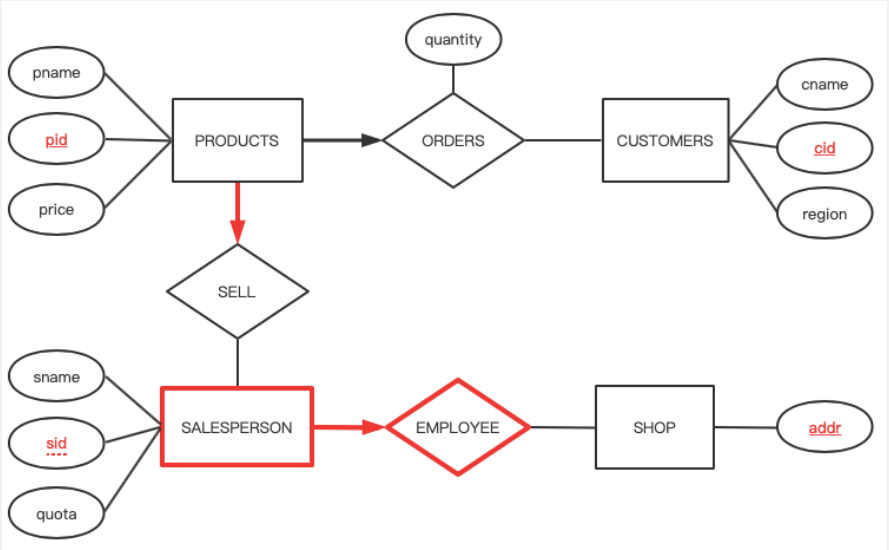
(c) **[4 points]** Now, let's complete the ER diagram with the tables given above and some new translations. You need to underline the primary keys, add arrows. If bolding a line/arrow, be sure to clearly make it bold. If here's weak entity, also be sure to clearly show it.

- Each product sold by exactly one salesperson.
- The sid of salesperson in the same shop is unique, however two salesperson in different shop might have the same sid.
- A salesperson can only work in one shop.
- The address can uniquely identify a shop.



Solution

2 bold arrow & 1 blod entity & 1 blod relation.



III Indexes and B+ Trees [9 points]

1. **[5 points]** Basics of B+ Trees.

   (a) **[1 point]** What is the maximum fan out F of an order d B+ tree?

   Solution
   The maximum fan out is the maximum occupancy plus one, or $2d + 1$, by definition.

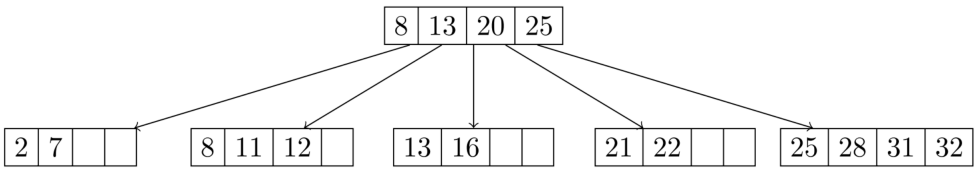
   (b) **[2 points]** Assume that each leaf can hold F data entries. What is the maximum number of data entries that an order d B+ tree of height 5 can store? Express your answer as a function of F. (Note that a height 1 B+ tree only has a root node).

   Solution
   We are assuming that the leaf pages hold $F = 2d + 1$. At height 1, the root node is a leaf node and therefore the index holds F. At height 2, there is a root node and its children are leaf nodes. The root node has a max fan out of F children and each child is a leaf node that can hold max F entries. Therefore the index can hold F^2 total entries. We are accepting $F^5, (2d + 1)^5$ and any other fifth power of fan out.

   (c) **[2 points]** Again, assume that each leaf can hold F data entries. And we have indexed a file with 1×10^9 records with this order d B+ tree. What is the minimum number of I/Os it will take to check if a data entry exists in this tree? Express your answer as a function of F.

   Solution
   $\lceil \log_F(\frac{10^9}{F}) \rceil$

2. **[4 points]** Consider the following B+ tree of order 2.



   (a) **[1 point]** How many nodes split when you insert 27?

   Solution

   - To insert 27, we follow the rightmost pointer in the tree from the root node because $27 > 8$, $27 > 13$, $27 > 20$, $27 > 25$.
   - We get to the leaf node containing (25, 28, 31, 32) and attempt to insert 27, but the leaf node is at maximum capacity and the insertion breaks the occupancy invariant. Therefore we must split the leaf into (25, 27) and (28, 32, 21) and copy up key 27 because this is a leaf node.
   - We attempt to insert 27 into the parent node containing (8, 13, 20, 25) but the root is also at maximum capacity. We split it into (8, 13) and (25, 28) pushing up key 20 because it is an inner node. Our final tree looks like and we split two nodes.

   (b) **[1 point]** After inserting 27 into the tree, you also insert 26. How many nodes split as a result of inserting 26?

   Solution

   - To insert 26, we first look at the root node and follow the right pointer because $26 \geq 20$. We then follow the middle pointer at the inner node below the root node because $26 > 25$ but $26 < 28$. We get to the leaf node containing (25, 27)
   - We attempt to insert 26 into the leaf node containing (25, 27) and since 3 is less than the

occupancy invariant 4 we insert 26 and are done. We did not split any nodes.

(c) **[2 points]** Assume that after inserting 26 and 27, you insert the keys 34, 35, 36, . . ., 100. After all these insertions, what keys are in the leftmost leaf node?

Solution
You should be able to identify that we are only inserting keys into the right side of the tree. This is similar to bulk loading in that the left leaf nodes will not be touched. Therefore, we know that the leftmost leaf node will not be modified and will contain (2, 7) after the insertions to the right side of the tree.

IV External Sorting [8 points]

1. **[2 points]** True or False:

   (a) Increasing the number of buffer pages don't affects the number of I/Os performed in Pass 0 of an external sort.

   Solution
   True: Regardless of the number of buffer pages, every pass of an external sort (including Pass 0) performs the same number of IOs.

   (b) Double buffering reduces the time it takes to sort records within a single page.

   Solution
   False: Double buffering allows a program to concurrently perform computations and fetch data from disk. Once a page of data is resident in memory, double buffering will not help speed up computation on it.

2. **[6 points]** Assume that each page is 4 KB large, and that you have a 24KB buffer pool (with 6 frames).

   (a) **[2 points]** How many passes would it take to externally sort an 512KB file? Include the initial sorting pass and subsequent merging passes in your answer. You need to simplify your answer.

   Solution
   3: The file contains 128 pages, and B = 6, and thus we need $\left(1 + \left\lceil \log_5 \left(\left\lceil \frac{128}{6} \right\rceil \right) \right\rceil = 3\right)$ passes.

   (b) **[2 points]** What would be the total cost in I/Os for this external sort?

   Solution
   768: 2∗128∗#passes = 256*3 = 768 (I/Os).

   (c) **[2 points]** What is the minimum number of additional buffer frames we require to reduce the number of passes (from part 1) by 1?

   Solution
   6: In order to sort the file in two passes, we need B buffer frames where $B(B-1) \geq 128$. The smallest such B is 12. Thus, the number of additional pages is $12 - 6 = 6$.

V Join Algorithms [8 points]

1. **[5 points]** Consider a relation R with attributes (x, y) and a relation S with attributes (y, z). Column y in S is a key and the set of values of y in R are the same as the set of values of y in S. Assume that there are no indexes available and that there are 25 pages in the buffer available. Table R is 1,500 pages with 50 tuples per page. Table S is 400 with 100 tuples per page. Compute the I/O costs for the following joins. Assume the simplest cost model, where pages are read and written one at a time.

   (a) **[1 point]** Block nested loops join with R as the outer relation and S as the inner relation.

   Solution
   $|R| + \lceil |R|/B \rceil \times |S| = 1500 + \lceil 1500/25 \rceil \times 400 = 25500$ OR
   $|R| + \lceil |R|/(B-1) \rceil \times |S| = 1500 + \lceil 1500/24 \rceil \times 400 = 26700$ OR
   $|R| + \lceil |R|/(B-2) \rceil \times |S| = 1500 + \lceil 1500/23 \rceil \times 400 = 27900$
   Since, these three equations all appears in some materials, any were acceptable.

   (b) **[1 point]** Block nested loops join with S as the outer relation and R as the inner relation.

   Solution
   12500: $|S| + \lceil |S|/B \rceil \times |R| = 400 + \lceil 400/25 \rceil \times 1500 = 24400$ OR
   $|S| + \lceil |S|/(B-1) \rceil \times |R| = 400 + \lceil 400/24 \rceil \times 1500 = 25900$ OR
   $|S| + \lceil |S|/(B-2) \rceil \times |R| = 400 + \lceil 400/23 \rceil \times 1500 = 27400$
   Since, these three equations all appears in some materials, any were acceptable.

   (c) **[1 point]** Sort merge join with R as the outer relation and S as the inner relation.

   Solution
   $2|\mathrm{R}| \left(1 + \lceil \log_{\mathrm{B}-1} \lceil |\mathrm{R}| / \mathrm{B} \rceil \rceil\right) + 2|\mathrm{S}| \left(1 + \lceil \log_{\mathrm{B}-1} \lceil |\mathrm{S}| / \mathrm{B} \rceil \rceil\right) + |\mathrm{R}| + |\mathrm{S}|$
   $= 2(1500)\,(1 + \log_{24} \lceil 1500/25 \rceil) + 2(400)\,(1 + \log_{24} \lceil 400/25 \rceil) + 1500 + 400 = 12500$
   We cannot use the cost model of 3(M+N) because we cannot sort the larger of the two relations in 2 passes.

   (d) **[1 point]** Sort merge join with S as the outer relation and R as the inner relation.

   Solution
   12500: Same as (c).

   (e) **[1 point]** Hash join with S as the outer relation and R as the inner relation.

   Solution
   5700: $3(\lceil R \rceil + \lceil S \rceil) = 3(1500 + 400) = 5700$.

2. **[3 points]** Consider a new case, i.e. B>4 pages worth of buffer space, and relations M and N of size > B. Please fill the blanks below with "always", "sometimes" or "never".

   (a) **[1 point]** Block nested loop join is _____ better than page-oriented nested loop join.

   Solution
   Always: Block nested loop join is page-oriented on steroids, and steroids are always good.

   (b) **[1 point]** Sort-merge join is _____ better than hash-join.

   Solution
   Sometimes: Hash join is cooler if one relation is really small and one is very large. Sort dominates if you have lots of duplicate join keys.

   (c) **[1 point]** Hybrid Hash-Join is _____ better than block-nested loops join.

   Solution
   Sometimes: Nested loops works for non-equijoins and hash does not. For equijoins, hash is often better since it only makes a small number of passes over each relation, whereas block nested-loops still may visit the inner relation many times. If one relation fits in memory, the two

algorithms are about equivalent.

VI Query Optimization [10 points]

Consider two relations Cat(age, weight, price) and Pocket(money), with 150 tuples and 100 tuples respectively. We have an index on Cat.age with 15 unique integer values uniformly distributed in the range [1, 15], an index on Cat.weight with 30 unique float values uniformly distributed in the range [2001, 5000], an index on Cat.price with 10 unique integer values uniformly distributed in the range [11, 20], and an index on Pocket.money with 15 unique integer values uniformly distributed in the range [11, 25].
Use selectivity estimation to estimate the number of tuples produced by the following queries.

(a) **[1 point]** SELECT * FROM Cat

Solution
150: Select all tuples.

(b) **[2 points]** SELECT * FROM Cat WHERE age $\geq$ 10

Solution
60:
Selectivity $= \frac{max-v}{max-min+1} + \frac{1}{\#\ distinct\ values} = \frac{5}{15} + \frac{1}{15} = \frac{2}{5}$,
$150 \times \frac{2}{5} = 60$.

(c) **[2 points]** SELECT * FROM Cat WHERE age $<$ 5 AND weight $\leq$ 3000

Solution
13:
Selectivity = Sel(age $<$ 5) $\times$ Sel(weight $\leq$ 3000) $= \frac{4}{15} \times \left(1 - \frac{2000}{3000}\right) = \frac{4}{45}$,
$150 \times \frac{4}{45} = 13.333 = 13$.

(d) **[2 points]** SELECT * FROM Cat WHERE age $>$ 10 OR price $\geq$ 15

Solution
110:
Selectivity = Sel(age $>$ 10) + Sel(price $\geq$ 15) - Sel(age $>$ 10) $\times$ Sel(price $\geq$ 15) $= \frac{5}{15} + \frac{6}{10} - \frac{5}{15} \times \frac{6}{10} = \frac{11}{15}$,
$150 \times \frac{11}{15} = 110$.

(e) **[3 points]** SELECT * FROM Cat, Pocket WHERE Cat.price = Pocket.money

Solution
1000:
Selectivity $= \frac{1}{\max(10,15)} = \frac{1}{15}$,
$150 \times 100 \times \frac{1}{15} = 1000$.

VII Transaction and Concurrency [10 points]

Consider the following schedule. (For each of the questions below, you may mark zero (ϕ), one or more than one of the choices.)

	T1	T2	T3	T4
1	R(A)			
2		R(A)		
3			R(C)	
4			W(C)	
5		R(B)		
6		W(B)		
7	R(B)			
8				R(B)
9	R(C)			
10				R(C)
11				W(B)
12		commit		
13			commit	
14	commit			
15				commit

(a) **[1 point]** What transactions is T1 pointing to in the conflict graph for the schedule above?
A. T1
B. T2
C. T3
D. T4

Solution
D

(b) **[1 point]** What transactions is T2 pointing to in the conflict graph for the schedule above?
A. T1
B. T2
C. T3
D. T4

Solution
A, D

(c) **[1 point]** What transactions is T3 pointing to in the conflict graph for the schedule above?
A. T1
B. T2
C. T3
D. T4

Solution
A, D

(d) **[1 point]** What transactions is T4 pointing to in the conflict graph for the schedule above?
A. T1
B. T2
C. T3
D. T4

Solution
None

(e) **[3 points]** Which of the following locking disciplines could have produced the above schedule?
A. 2 phase locking
B. Strict 2 phase locking

Solution
A.
2PL is possible because each transaction could release it's lock immediately after it's done with its operation and so this schedule would be allowed to happen. Strict 2PL is not possible because transactions have to hold their locks (ex. T1 would not be able to R(B) in 7 because T2 would have to hold its exclusive lock on B until it ends)

(f) **[3 points]** Which of the following schedules below are conflict equivalent to the schedule above?
A. T3, T1, T2, T4
B. T2, T3, T1, T4
C. T4, T3, T1, T2
D. T1, T2, T3, T4
E. T3, T2, T1, T4

Solution
B, E.
Just by doing a topological sort, we know that T4 has to be last and T1 has to be second to last. Whether T2 or T3 comes first is irrelevant since they have no incoming edges, so we have 2 possible conflict equivalent schedules.

VIII Logging and Recovery [13 points]

1. **[5 points] General Logging and Recovery.**
   Mark the boxes for all true statement(s):

   (a) Schedules produced by two phase locking are guaranteed to prevent cascading aborts.

   Solution
   F: Strict 2PL is needed to guarantee this.

   (b) Strict two phase locking is both necessary and sufficient to guarantee conflict serializability.

   Solution
   F: Sufficient but not necessary.

   (c) In a system that uses strict two-phase locking, if a transaction aborts, it releases all of its locks as soon as rollback is complete.

   Solution
   T.

   (d) In a system that uses strict two-phase locking, a transaction that only performs reads can never enter a deadlock cycle.

   Solution
   F.

   (e) When aborting a transaction, it is necessary to modify pages on disk.

   Solution
   T. steal policy.

   (f) During recovery, the ARIES protocol redo aborted transactions.

   Solution
   T.

   (g) When a transaction commits, any modified buffer pages must be written to durable storage.

   Solution
   F: no force policy.

   (h) In ARIES recovery, after the analysis phase, the recLSN of each page in the dirty page table must be larger than the pageLSN of the corresponding page.

   Solution
   F: The page could have been updated and flushed from the buffer pool between the last checkpoint and time of crash. The flushed page would have a pageLSN from its most recent update, which is after the recLSN in the checkpoint.

   (i) If PageLSN is greater than the max LSN flushed so far (flushedLSN), we can safely write this page to disk.

   Solution
   T: WAL requires that.

   (j) Write-Ahead Logging (WAL) guarantees that a transactions log records are flushed to disk before the transaction commit.

   Solution
   T.

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	Begin Checkpoint	-
90	update: T1 writes P3	70
100	End Checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	Null
180	CLR: undo T4 LSN 130	150

Transaction table at time of checkpoint

Transaction ID	lastLSN	Status
T1	70	Running
T2	60	Running
T3	30	Running
T4	50	Running

Dirty page table at time of checkpoint

Page ID	recLSN
P5	50
P1	40

2. **[8 points] Recovery.**
Your database server has just crashed due to a power outage. You boot it back up, find the following log and checkpoint information on disk, and begin the recovery process. Assume we use a STEAL/NO FORCE recovery policy.

(a) **[2 points]** At the end of the Analysis phase, what transactions will be in the transaction table, and with what lastLSN and Status values?

Solution

Transaction ID	lastLSN	Status
T1	90	Running
T3	30	Running
T4	180	Aborting
T5	160	Running

We also accept the answer by moving the status of transactions in the above table from "Running" to "Aborting".

(b) **[2 points]** At the end of the Analysis phase, what pages will be in the dirty page table, and with what recLSN values?

Solution

Page ID	recLSN
P1	40
P2	160
P3	90
P5	50

(c) **[4 points]** At which LSN in the log should redo begin? Which log records will be redone (list their LSNs)?

Solution

LSN 40: Redo should begin at LSN 40, the smallest of the recLSNs in the dirty page table. The following log records should be redone:

$$40, 50, 60, 80, 90, 100, 110, 120, 130, 140, 150, 160, 180.$$

30 is skipped because it precedes LSN 40. 70 is skipped because P2.recLSN $= 160 > 70$. Entries that are not updates are skipped. The CLR record is not skipped, nor is the LSN that it undoes. We also accept the answer:

$$40, 50, 60, 90, 110, 130, 160, 180,$$

where only update and CLR logs are remained.

IX Data Mining and Machine Learning [20 points]

1. **[4 points] General Data Mining and Machine Learning.**
   Mark the boxes for all true statement(s):

   (a) In multi-dimensional data model, it contains one fact table and multiple dimension tables.

   Solution
   True.

   (b) The order of the basic KDD (Knowledge Discovery in Database) process is Data Selection, Data Cleaning, Evaluation, Data Mining & ML.

   Solution
   False: Data Mining comes before Evaluation.

   (c) In supervised machine learning labeled data is used to train a model.

   Solution
   True: The labeled data is the "supervision" in the name.

   (d) Classification models would be a good candidate when trying to predict the total amount of time a user spends browsing a page.

   Solution
   False: Classification produces a discrete label (like "True/False"); to get a numeric value like an amount of time you might want to use regression.

   (e) The ultimate goal in machine learning is to find a model that best fits the training data.

   Solution
   False: Fitting the training data is a means to the end goal of predicting unknown information. As we learned, you can overfit the training data if you make that your ultimate goal.

   (f) The k-means algorithm is guaranteed to converge to the global optimum.

   Solution
   False: k-means can produce a sub-optimal answer, and it sensitive to the initial position of the centers; it does compute what's called a "local optimum".

   (g) The bag-of-words model encodes text as a vector.

   Solution
   True.

   (h) A common form of feature engineering on continuous data is one-hot-encoding.

   Solution
   False: One-hot encoding is used for capturing the presence of boolean features.

2. **[8 points] K-Means.**

   (a) **[2 points]** True or False:
   Suppose we are going to cluster the following dataset: $\{(x_1, y_1), ..., (x_n, y_n)\}$, denote $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, we can have a decentralized dataset $\{(x'_1, y'_1), ..., (x'_n, y'_n)\}$ where $x'_i = x_i - \bar{x}$, $y'_i = y_i - \bar{y}$, $i = 1, 2, ..., n$. The k-means algorithm will converge to the same result if we choose the initial centers as $\{(x_s, y_s), (x_t, y_t)\}$ for the original dataset and $\{(x'_s, y'_s), (x'_t, y'_t)\}$ for the decentralized dataset, where s, t are two different constants.

Solution
True

(b) **[2 points]** Which of the following can act as possible termination conditions in k-means? (you may mark zero (ϕ), one or more than one of the choices.)
A. For a fixed number of iterations.
B. Assignment of observations to clusters does not change between iterations. Except for cases with a bad local minimum.
C. Centroids do not change between successive iterations.
D. Terminate when RSS falls below a threshold.

Solution
A, B, C, D.
All four conditions can be used as possible termination condition in K-Means clustering:
This condition limits the runtime of the clustering algorithm, but in some cases the quality of the clustering will be poor because of an insufficient number of iterations. Except for cases with a bad local minimum, this produces a good clustering, but runtimes may be unacceptably long. This also ensures that the algorithm has converged at the minima. Terminate when RSS falls below a threshold. This criterion ensures that the clustering is of a desired quality after termination. Practically, it's a good practice to combine it with a bound on the number of iterations to guarantee termination.

(c) **[2 points]** In which of the following cases will k-means clustering fail to give good results? (you may mark zero (ϕ), one or more than one of the choices.)
A. Data points with outliers.
B. Data points with different densities.
C. Data points with round shapes.
D. Data points with non-convex shapes.

Solution
A, B, D.
K-Means clustering algorithm fails to give good results when the data contains outliers, the density spread of data points across the data space is different and the data points follow non-convex shapes.

(d) **[2 points]** The following is a set of one-dimensional points: $\{-4, 0, 1, 2, 3, 5, 7, 10, 13, 22\}$, perform two iterations of k-means on these points using the two initial centroids 1 and 7.
1) What are the two new centroids after the first iteration?
2) What are the two new centroids after the second iteration?

Solution
first iteration: 2/5; 57/5
second iteration: 7/6; 13

3. **[8 points] Linear Regression.**
Given a set of i.i.d. data points $(x_1, y_1) \cdots (x_n, y_n)$, where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ denote the input feature and the output response, respectively. By assuming the linear model is a reasonable approximation, we consider fitting the model via least squares approaches, in which we choose coefficients θ and θ_0 to minimize the Residual Sum of Squares (RSS),

$$\hat{\theta},\ \hat{\theta}_0 = \underset{\theta,\ \theta_0}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \theta x_i - \theta_0)^2. \tag{1}$$

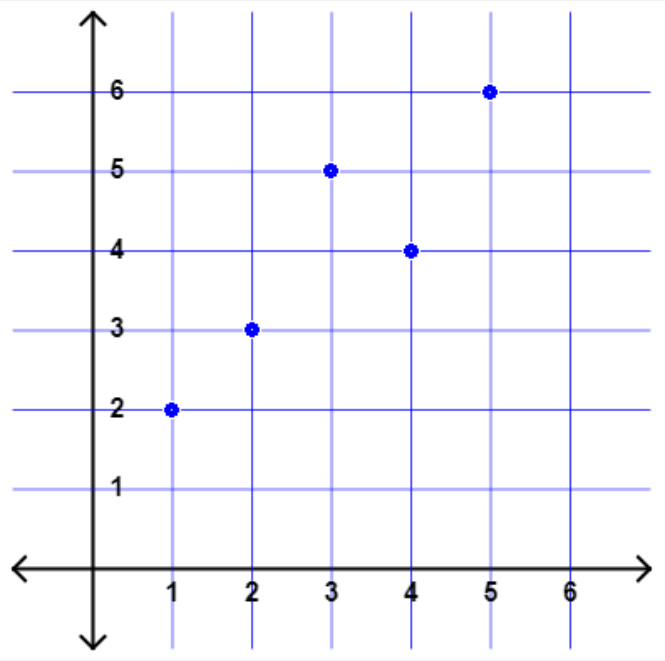
Based on 5 observations of (x, y):

$$(1, 2), \quad (2, 3), \quad (3, 5), \quad (4, 4), \quad (5, 6), \tag{2}$$

please answer the following questions:

(a) **[1 point]** Draw a 6 by 6 space with all the 5 data points.

Solution



(b) **[3 points]** Estimate the model parameters θ and θ_0.

Solution
$\hat{\theta} = \frac{9}{10}$, $\hat{\theta}_0 = \frac{13}{10}$:
By setting the derivative of the objective function w.r.t. θ and θ_0 equal to 0, we have

$$\hat{\theta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \tag{3}$$

$$\hat{\theta}_0 = \bar{y} - \hat{\theta}\bar{x}, \tag{4}$$

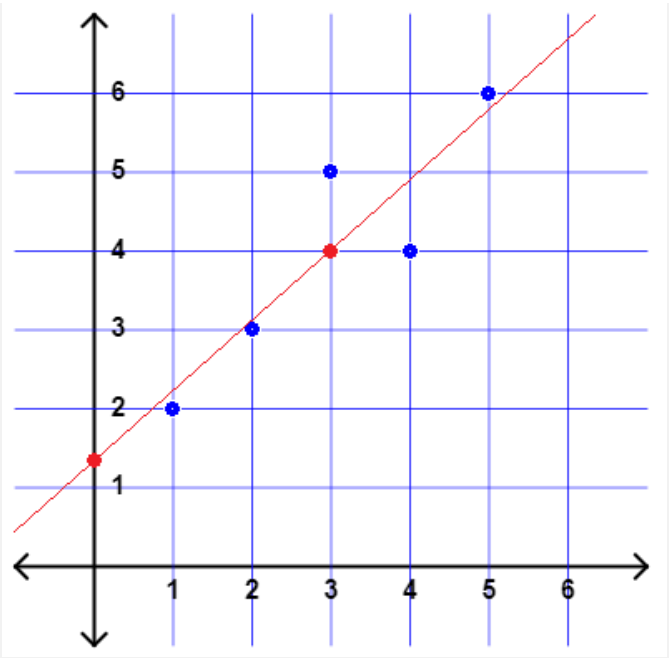
where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = 3$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = 4$. Then, substituting the 5 data points into the above solutions, immediately leads to our solution.

(c) **[2 points]** Using (1), argue that the least squares line,

$$y = \hat{\theta}x + \hat{\theta}_0, \tag{5}$$

always passes through the points $(0, \hat{\theta}_0)$ and $(\bar{x}, \bar{y})$, where $\bar{x} = \frac{1}{5}\sum_{i=1}^{5} x_i$ and $\bar{y} = \frac{1}{5}\sum_{i=1}^{5} y_i$, and plot the line in the picture of (a).

Solution
Because $(0, \hat{\theta}_0) = (0, \frac{13}{10})$ and $(\bar{x}, \bar{y}) = (3, 4)$ satisfy the learned least squares line $y = \frac{9}{10}x + \frac{13}{10}$. Therefore, these two points determine the our line, and we can plot it in the following figure.



(d) **[2 points]** Find the data point which makes the highest contribution to RSS, and show the geometric interpretation in the picture of (a).

Solution

$(3, 5)$:

The contribution to RSS is measured by $\left(y_i - \left(\hat{\theta} x_i + \hat{\theta_0}\right)\right)^2 = \left(y_i - \left(\frac{9}{10} x_i + \frac{13}{10}\right)\right)^2$, and the highest contribution $\left(y_i - \left(\frac{9}{10} x_i + \frac{13}{10}\right)\right)^2 = (5-4)^2 = 1$ is made by the point $(3, 5)$.
It actually indicates the area of the red square showed in the figure below.