U88387934 Zizi Tan

Boston University
CS 562- Advanced Database Systems

Take Home Midterm Exam – Spring 2020
Return by: 9:00AM EST, Saturday, April 11, 2020 on Gradescope

Instructions

- This exam is OPEN books and notes.
- You have **24 hours** to complete it. There are 4 Problems.
- Make sure that you answer all the questions.
- Use the Gradescope and return your answer by 9:00AM EST, Saturday, April 11, 2020.

Good Luck!

Problem 1 - Spatial Databases [Points: 40]

1. Consider the point P = (15, 65). Give the Z-value (decimal number) of P if we use K=3 bits per dimension. The address space is the square $[0,80)^2$ (i.e. values range between 0 and 80 in each dimension.)

2. Consider again the data space $[0, 80)^2$ and a Range (Window) query R with a low-left corner (5,6) and upper-right corner (15,24). Assume again that the K = 3. Find the equivalent 1D ranges using the Z-values that this query is mapped to. That is, find the set of 1D ranges that this query is mapped using the Z-ordering curve.

3. Consider the following point set S = {a = (1,1), b = (2,4), c = (3,6), d = (5,5), e = (6,7), f = (9,4)}. Create a kd-tree for S. Using the kd-tree that you created, answer the 1-NN query for the query point Q = (5, 4). Report the answer and which nodes you have to visit. You need to report ALL the nodes of the kd-tree that you have to visit in order to answer the query.

4. Suppose that you want to use a Space Filling Curve (SFC) in order to improve the performance and/or the quality of the insertion and the split functions of the R-tree. Please, present an approach that uses an SFC to implement the **Insert** and the **SplitNode** functions. Which SFC you will use (e.g., which one of the Z-value, Hilbert-value, other) and why?

Problem 1

1. P(15, 65) z-value 01 11

space $[0, 80)^2$ 00 10

$k = 3$ z

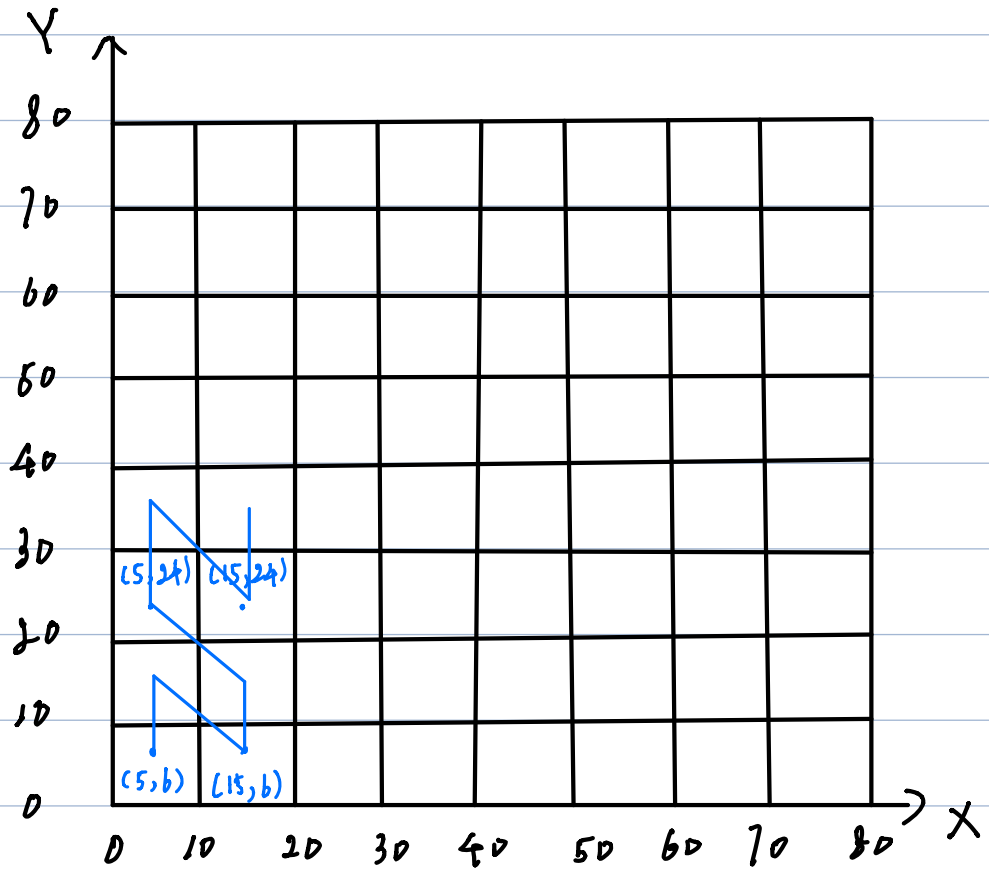
$80 \div 2 = 40$ $15 < 40$, $65 > 40$ 01

$\vdots$ $15 < 20$, $65 > 60$ 01

$15 > 10$, $65 < 70$ 10

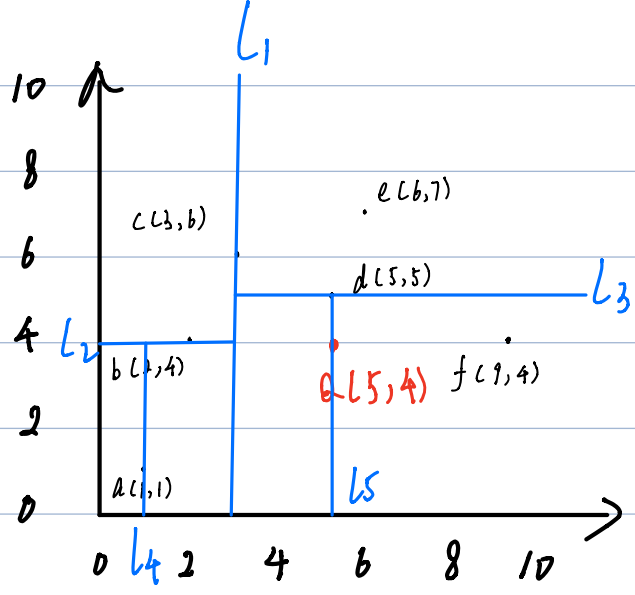
∴ The z value is 01 01 10.

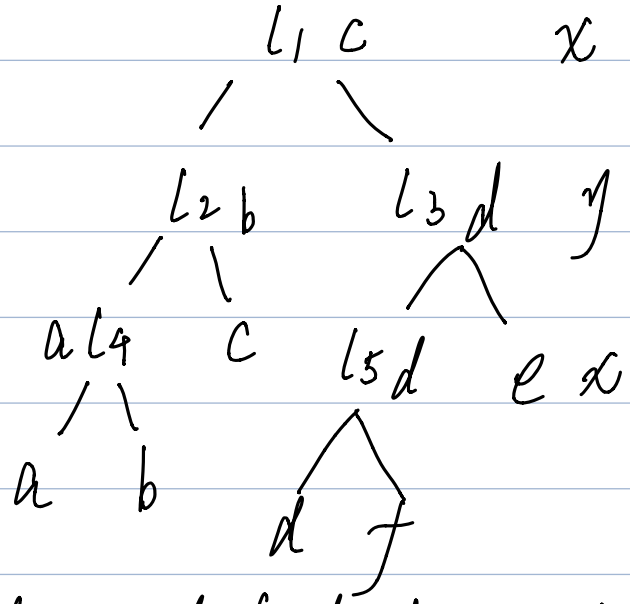
2.



Range [0, 4] and [6]

3.





Firstly, search the kd tree and find the minimum sub-region where the point Q Locates. In this case, we need to find L_5. We will visit L_1, L_3 and L_5.

Then, we will get d and f. Point d is closer. Therefore, we marked it as the current nearest neighbor and the distance $d_{min} = 1$.

Then, we do backtracking on the tree.
Backtrack to point e and the distance to e is even Longer. Discard e.
Backtrack to L_1 and the distance to L_1 region is 2, which is bigger than d_{min}. Then, we prune the left subtree of L_1.

Therefore, the answer is point d. We need to visit L_1, L_3, L_5, d, f, e.

4. Use Hilbert value.

A paper from Faloutsos, Symposium on Principles of Database System shows Hilbert value performs better.

Implementation:

The Hilbert value of a rectangle/MBR is defined as the Hilbert value of its center.

For every node N of the tree, we store its MBR and the Largest Hilbert Value of the data rectangles that belong to the subtree with root N.

Insertion:

To insert a reatangle r with hilbert value as a key, in each level, we choose the node with minimum LHV among the siblings. When a leaf node is reached, r is inserted in its correct order according to h.

SplitNode:

We treat the overflowing nodes either by moving some of the entries to one of the s-1 cooperating siblings or by splitting s nodes into s+1 nodes.

Problem 2 - Spatial & Spatio-temporal Databases [Points: 20]

Consider the following dataset $\mathcal{D}$ of 14 objects in a 2-dimensional space (12 points and 2 rectangles): a=(1,3), b=(1,4), c=(1,7), d=(2,5), e=(2,8), f=(3,5), g=(3,1), h=(5,1), i=(6,1), j=(6,7), k=(7,0), l=(7,1), R1={ (4,2), (5,3)}, R2={ (7,5), (8,6) }.

1. Create an R-tree on $\mathcal{D}$. Assume that the maximum number of objects per page (M) is 4 and the minimum (m) 2. Show all the nodes and the entries of each node. (You can create the R-tree anyway you want. The only requirement is that the tree is correct.)

2. Using the tree created above, show how to answer (i) the range query Q={(4,3), (8,6)} and (ii) the nearest neighbor query NNQ={(6,3)}. Report the nodes visited and the answer for each query.

3. Consider a Time Parameterized nearest neighbor (TP NN) query on the same dataset. The query point is Q= [(2, 1)] and is moving with velocity (vx, vy) = (0, 1). Report the TP answer to this query and the nodes that you will visit in the R-tree that you created above in order to answer the query.

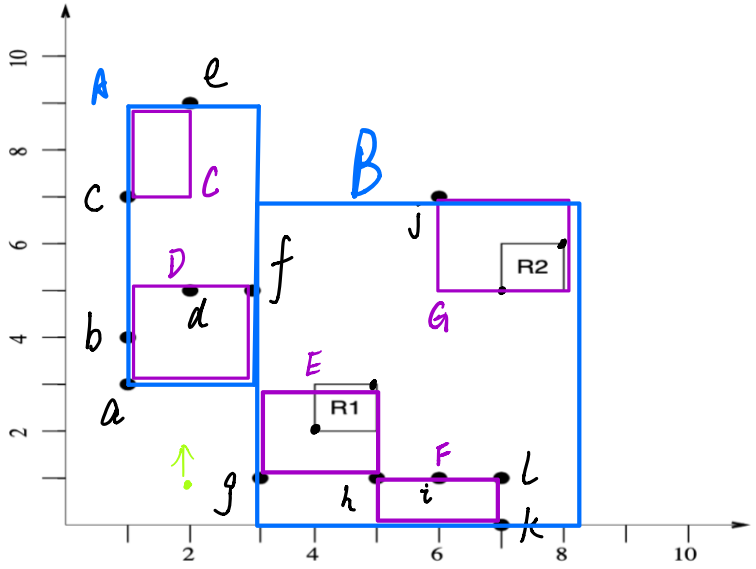
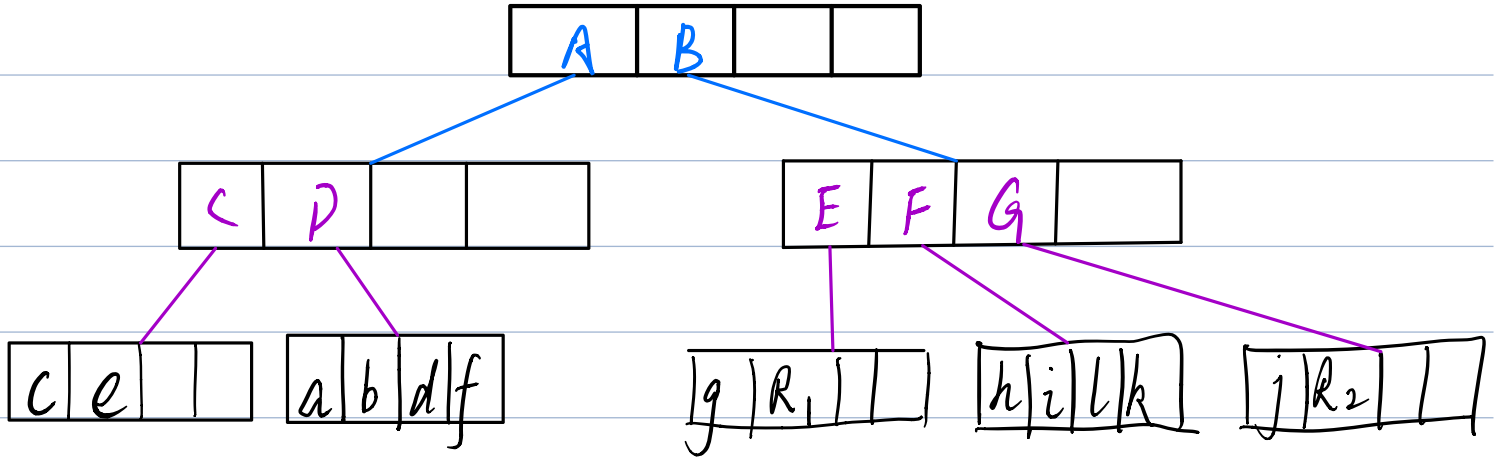


Figure 1: The dataset $\mathcal{D}$.

1. R tree



2. Range Query. $Q = \{(4,3), (8,1)\}$

Report: B, E, g, R_1, F, G, j, R_2

Answer: R_2, R_1

Notice: we also need to check the objects which lie exactly on the bounds of the query region.

Nearest Neighbor $Q = \{6,3\}$

Priority Queue state:

1. PQ: B, A
2. PQ: E, F, G, A
3. PQ: R_1, g, F, G, A

Report: B, A, E, F, G, R_1, g

Answer: R_1

3.

TP Query

Result	Expiry Time	Change	Nodes that will be visited
{g}	0	{g}	B, A, E, F, G, g, R_1
{g, a}	1	{a}	A, B, C, D, a, b, d, f
{a}	2	{-g}	A, B, C, D, a, b, d, f
{a, b}	2.5	{b}	A, B, C, D, a, b, d, f
{b, d}	3	{d, -a}	A, B, C, D, a, b, d, f
{d}	4	{-b}	A, B, C, D, a, b, d, f
{c}	6	{c, -d}	A, B, C, D, c, e
{e}	7	{e, -c}	A, B, C, D, c, e

Problem 3 – Temporal Databases [Points 30]

1. Consider the following temporal evolution of a dataset S:

(time, oid, operation)

(1, 1, i)	(11, 11, i)	(21, 21, i)
(2, 2, i)	(12, 12, i)	(22, 22, i)
(3, 3, i)	(13, 5, d)	(23, 23, i)
(4, 4, i)	(14, 7, d)	(24, 12, d)
(5, 5, i)	(15, 8, d)	(25, 25, i)
(6, 6, i)	(16, 16, i)	
(7, 7, i)	(17, 11, d)	
(8, 8, i)	(18, 10, d)	
(9, 9, i)	(19, 19, i)	
(10, 10, i)	(20, 20, i)	

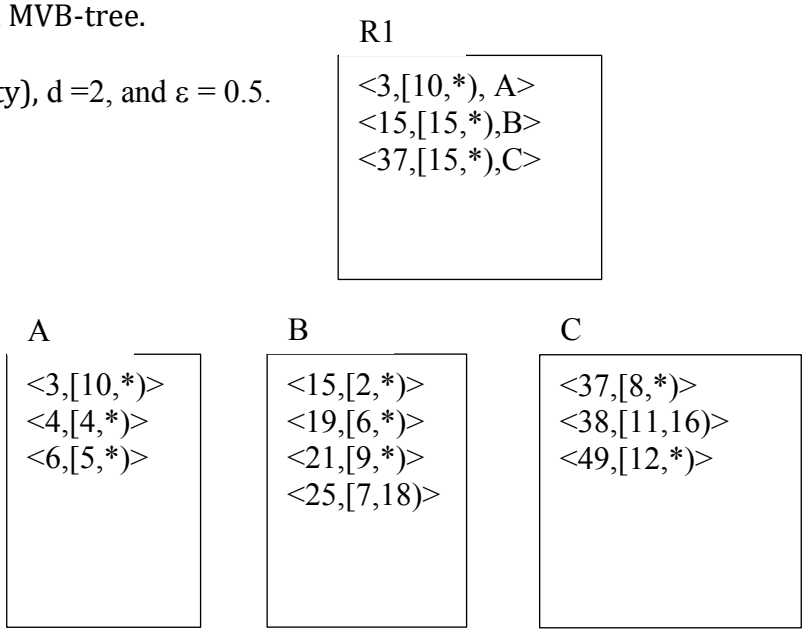
Assume that you create a Snapshot Index on the evolution of S using blocks (pages) with capacity B=4 and usefulness parameter u=0.5.

Show the Access Forest of the Snapshot Index, after the execution of the last operation. Show how to answer the timeslice query $t_q = 21$ using the snapshot index that you created. (Report all the nodes that you have to visit).

2. Consider the All-versions query: The query provides an object id (oid or key) and the answer is all the versions of the given oid (key). Discuss how to extend the structures of the Snapshot Index (including the entries, if needed) in order to answer this query efficiently. Provide the cost of answering the query in terms of number of I/Os.

3. Consider the following state of a MVB-tree.

We assume that b=6 (block capacity), d =2, and $\varepsilon = 0.5$.



(Notice that other nodes may exist but they are not relevant to the problem and therefore they are ignored).

Show the state of the MVB-tree after all the following operations are performed:

Insert 30 at time 20, **insert** 28 at time 21, **insert** 16 at time 22, **delete** 37 at time 23.

Problem 4- Time Series (Basic) [Points 10]

1) 1. Consider the following time series X = [1, 2, 2, 3, 5] and Y = [3, 1, 1, 6, 7]. Compute the DTW distance between X and Y . You can use the L_1 distance between the i-th element of X and the j-th element of Y when you do your computation. Show the matrix and the final result.
2) What is the relationship between the Euclidean distance and the DTW distance for time series?

Problem 3.

1.

Block A
<1, [1, *]>
<2, [2, *]>
<3, [3, *]>
<4, [4, *]>

Block B
<5, [5, 13]>
<6, [6, 15]>
<7, [7, 14]>
<8, [8, 15]>

Block C
<9, [9, 24]>
<10, [10, 18]>
<11, [11, 17]>
<12, [12, 24]>

Block D
<6, [15, *]>
<16, [16, *]>
<19, [19, *]>
<20, [20, *]>

Block E
<21, [21, *]>
<22, [22, *]>
<23, [23, *]>
<9, [24, *]>

Block F
<25, [25, *]>

AT	time	Block
	0	S
	1	A
	5	B
	9	C
	15	D
	21	E
	25	F

Access forest:

S[0, *] ↔ A[1, *] ↔ D[15, *] ↔ E[21, *] ↔ F[25, *]

A[1, *] ↔ B[5, 15] ↔ C[9, 24]

tq = 21 Find Block E from AT array.
start from E and search the forest.

Report: E → D → A → C

2. Use an additional hashing function to track the key, the corresponding blocks which include the copy and deletion procedure.

hash (key) → Block 1 $\xrightarrow[\text{when the block dies}]{\text{copy}}$ Block 2 $\xrightarrow[\text{the key}]{\text{delete}}$ Marked as deleted $\xrightarrow[\text{this key}]{\text{create}}$ Block 3.

In the copy, deletion and creation procedure, we can just use a pointer to track it down.

That can be addressed in $O(\log_b n + v)$ I/O, where v is the number of versions.

3. $b = 6$, $d = 2$, $\varepsilon = 0.5$

$b = kd \Rightarrow k = 3$

Weak version condition : $d = 2$

Strong version condition : $(1+\varepsilon)d \le N \le (k-\varepsilon)d$

$3 \le N \le 5$

R_1

$\langle 3, [10, *), A \rangle$
$\langle 15, [15, 22), B \rangle$
$\langle 37, [15, 23), C \rangle$
$\langle 15, [22, *), D \rangle$
$\langle 21, [22, *), E \rangle$

A

$\langle 3, [10, *) \rangle$
$\langle 4, [4, *) \rangle$
$\langle 6, [5, *) \rangle$

B

$\langle 15, [2, *) \rangle$
$\langle 19, [6, *) \rangle$
$\langle 21, [9, *) \rangle$
$\langle 25, [7, 18) \rangle$
$\langle 30, [20, *) \rangle$
$\langle 28, [21, *) \rangle$

C

$\langle 37, [8, 23) \rangle$
$\langle 38, [11, 16) \rangle$
$\langle 49, [12, *) \rangle$

D

$\langle 15, [2, *) \rangle$
$\langle 16, [22, *) \rangle$
$\langle 19, [6, *) \rangle$

E

$\langle 21, [9, *) \rangle$
$\langle 28, [21, *) \rangle$
$\langle 30, [20, *) \rangle$
$\langle 49, [12, *) \rangle$

Problem 4

(1)

5	6	8	10	7	9
3	4	6	6	7	11
2	4	4	4	7	11
2	3	3	3	6	11
1	2	2	2	7	13
	3	1	1	6	7

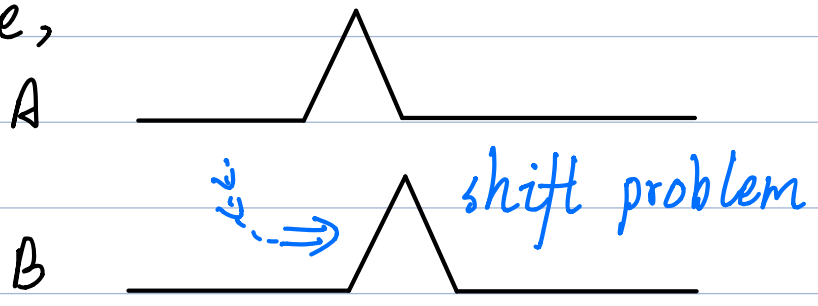
DTW = 9

(2) Euclidean distance is great than or equal to DTW distance.

In this case, Euclidean distance is $2+1+1+3+2=9$, equal to DTW.

Euclidean distance aligns the i-th point of time sequence A and i-th point of time sequence B, which would produce a really poor similarity score.

For instance,



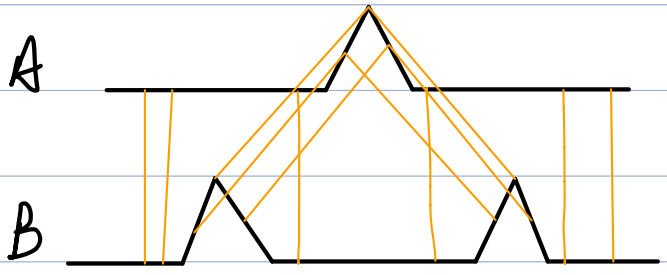
you will get poor similarity score.

However, the advantage is that the time complexity is O(N) where N is the length of the time series.

DTW can align different points in the other sequence, which is able to avoid the shift problem we mention above.

Disadvantage: DTW may consider two sequences similar when actually they are not.

For instance,



We need to apply warping window to avoid this problem to some degrees