**Big Data Analytics**

**Name:** Sreenath Reddy Kurukunda

**Student id:** 811211580

In this homework, there are 5 questions + 1 bonus question, covering the Z-order and R-tree family. If you can answer the bonus question correctly, you can obtain 20 extra points. The maximum mark for this homework is **120 points**, which will be later scaled.

1. Please list at least 2 differences between B+-tree and R-tree. [10 points]  
**ANSWER:-**

• The information is stored in leaf nodes in both the B+-tree and the R tree. The B+ tree can only store one dimension of data, whereas the R tree can store the multiple data dimensions.

• The B+ tree utilizes simple queue base level ordering, whereas R-tree doesn't have to use single queue-based level.

• The time complexity of the B+ tree is O log(n), which is used to access the keys at some range. The R tree, on the other hand, has a time complexity of O log(n) and is used to access the keys in N-dimensional ranges.

• The B+ tree can identify keys that fall within a specific range. The R tree can recognize all close values in two dimensions or more.

2. For Z-ordering curve, with 16 × 16 pixels (or cells), please use the bit-shuffling method to find the mappings between cell locations and Z-values below. [20 points]

2a. Please illustrate how to calculate the Z-value for the cell with the 2D location (6, 15) ***[note: cell locations start from 0 on each dimension]***. [10 points]  
**ANSWER:**

Consider the Binary numbers 6 and 15, which are written as

6 - 0110   
15 – 1111

Bit shuffling, we would then consider the two parameters (a1, a2, a3, a4) and (b1, b2, b3, b4) When these parameters have been shuffled, the resulting values are as follows: (a1, b1, a2, b2, a3, b3, a4, b4)

Similarly, the value for the 6 and 15 will be shuffled as follows.

(0 1 1 1 1 1 0 1)

2 Decimal Value = 20\*1 + 21\*0 + 22\*1 + 23\*1 + 24\*1 + 25\*1 + 26\*1 + 27\*0

= 1 + 0 + 4 + 8 + 16 + 32 + 64 + 0

= 125

As a result, the decimal value corresponding to cell (6,15) = 125.

2b. Given a Z-value 101 (decimal number), please identify the 2D location of the cell corresponding to this Z-value. [10 points]  
**ANSWER:**

01100101 is the binary equivalent of a given Z-value, 101. After bit shuffling, the Z-value is obtained.

Take this example: the given value is (a1, b1, a2, b2, a3, b3, a4, b4), and when the shuffling is reversed, the values are (a1, a2, a3, a4) and (b1, b2, b3, b4).

Similarly, the values for the given Z-value (0, 1, 1, 0, 0, 1, 0, 1) will be (0, 1, 0, 0) and (1, 0, 1, 1), which are (0100) - 4 (1011) - 11

As a result, the 2D location of the Z-value 101 is (4,11).

3. Given two 4-dimensional *minimum bounding rectangle* (MBRs) in an R-tree, *A* = (2, 10; 40, 60; 32, 40; -2, 11) and *B* = (3, 4; 30, 50; 30, 40; 20, 23), please use an MBR, *E* to minimally bound both MBRs *A* and *B*. [20 points]

*E* =

**ANSWER:**

Given that the smallest bounding rectangle of two 4-dimensional dimensions in the R tree is A = (2, 10; 40, 60; 32, 40; -2, 11) and

B = (3, 4; 30, 50; 30, 40; 20, 23)

Now, compare every one of the coordinates between A and B as follows:

for comparing A (2,10) and B (3,4) gives (2,10)

For (40,60) and (30,60) gives (30,60)

For (32, 40) and (30, 40) gets (30, 40).

For (-2, 11) and (20, 23) gets (-2, 23).

Therefore, E= is the minimum bonding rectangle of A and B in four dimensions (2,10; 30,60; 30,40; -2,23)

4. Given two *d*-dimensional MBRs, *E*1 = (x1min, x1max; x2min, x2max; …; xdmin, xdmax) and *E*2 = (y1min, y1max; y2min, y2max; …; ydmin, ydmax) (note: both E1 and E2 are represented by an array in the code), please give the pseudo code to check whether or not two MBRs are intersecting with each other. If yes, please return true; otherwise, return false. [20 points]

**ANSWER:**

To check whether two MBRs intersect with each other or not can be written as follows,

// let us consider that two 2-D arrays are store in two d-dimensional MBRs

E1 = (x1min, x1max; x2min, x2max; …; xdmin, xdmax)

E2 = (y1min, y1max; y2min, y2max; …; ydmin, ydmax)

// Let us consider a Boolean variable after checking whether it return true or false

Boolean is\_intersecting;

for (run until the last index)

{

If ((E1[i]>=E2[i] && E1[i]<=E2[i+1]) OR (E1[i+1]>=E2[i] && E1[i+1] <= E2[i+1]) OR (E2[i]>=E1[i] && E2[i]<=E1[i+1]) OR (E2[i+1]>=E1[i] && E2[i+1]<= E1[i+1]) )

{

                  is\_intersecting = true;

                  return is\_intersecting;

}

Else

{

                  is\_intersecting = false;

                  return is\_intersecting;

}

5. Write an algorithm to perform a range query (with rectangular shape, *Q*) in a quad-tree (please write comments to explain the meaning of your pseudo code). [30 points]

**ANSWER:**

For performing a range query in a quad-tree can be written as follows,

Array inRange[n];

// Let use a loop to run through all the values

Loop(I = 0 to max\_val)

{

       //let use the conditional statement to check if the values are in the range

       If( min\_range < I && I < max\_range)

       {

              // Then add the value to the array inRange

              inRange[j++] = I;

       }

}

// Now look for the child nodes in the quadtree

If(nodes = null)

{

       Return inRange[];

}

Else

{

       //Add all the node value to the array inRange[]

       For (I from 0 to n)

       {

              inRange[j++] = I;

       }

}

**Bonus Question [20 extra points]**

6. Read Sections 1 and 2 of the following paper and write a short survey about the formal definition of *reverse nearest neighbor* (RNN) query and solutions of RNN processing (including KM, YL, MVZ, SAA, and SFT). Use one paragraph for each solution (*note: please use your own words to describe the problem definition and solutions;* ***DO NOT*** *copy any sentences from the paper*).

Yufei Tao, Dimitris Papadias, and Xiang Lian. Reverse *k*NN Search in Arbitrary Dimensionality. In *Proceedings of the Very Large Data Bases Conference* (VLDB'04), pages 744-755, Toronto, Canada, Aug. 30-Sept. 3, 2004. *Located in the Library Course Reserves on the left-hand course menu.*

**ANSWER:**

RNN: The RNN was only used to create 2- dimensional datasets and only could give approximate values. The RNN methods are always being improved day by day, new analysis of the multi-dimensional data sets.

In RNN they are two kinds of methods,

1. The dataset of the nearest neighbor is all pre-computed.
2. The Dynamic values for all the nearest neighbors are considered.

KM: The neighbors are considered as the RNN algorithm. To solve that problem, Korn and Muthu Krishna developed the KM algorithm, which is used to obtain RNN by utilizing all the point’s nearest in the database. Then the values are kept in the R-tree, this is called as the RNN-tree.

YN: KM algorithm stores the dataset using the R-tree. It is very difficult because they are several data structures. It takes lot of time to get through every tree to solve this problem, Yang and Lin developed a method which is used to storing the data in the leaf node of the same tree rather than in a separate data structure. It makes very easy to move the tree.

MVZ: This MVZ algorithm can be used only with 2D datasets. This algorithm which are the nearest neighbors to the data are used for the pre-processing.

SAA: The static data is used with all the algorithm, which are including KM, YN, MVZ. None of the above algorithm can use the dynamic data. To support or use the dynamic data, the SAA algorithm is introduced. The dynamic data is processed in the two different phases.

1. The nearest neighbors are computed.
2. The data generated in the first phase are subjected to a nearest neighbor query.

SFT: This algorithm is the method uses multistep process to compute the nearest neighbor for the multi- dimensional dataset.

This methos has an issue that, can may result in negative misses, which may reduce by performing the query in huge dataset.

## Submitting Your Assignment

*All work must be your own. Copying other people’s work or from the Internet is a form of plagiarism and will be prosecuted as such.*

You may submit a Microsoft Word (.docx) document as an attachment. If you attach a document for your assignment, be sure to include your name in the text of the document and in the name of the document.

You can submit multiple times and only the last submission attempt will be considered for grading.

* Submissions sent by email will NOT be accepted.