COMP 6521

Advanced Database Technology and Applications

Project Report on

Lab Assignment 2

Course Instructor

Dr. Nematollaah Shiri

Submitted by

Sai Teja Prasadam	**40041931**
Prashanth reddy Somireddy	**40047831**
Bharath Parthasarathy	**40036730**

How to run the program:

Run the Runner.java from the file that loads the dataset and runs the queries.

Program description:

The program takes the input data set T and then processes through the first phase of the algorithm. In this phase, it forms the KD tree with the given point format (X, Y,Z). For forming this tree, for the initial level, the program reads all the datasets and sorts with coordinate 'X' and then takes the mean of it and places it as the tree root. In the next level, all the left values of the sorted tree are taken as the new dataset and sorts with 'Y' and takes the mean of that value places it at the root of the left list. The same thing for the right list. In the next level, we will have 4 datasets all these are sorted with 'Z' and takes the mean for each datasets and places them at the root of the tree. This cycles through until all the datasets are added to the KD tree.

The calculations for Total numbers of blocks, number of runs in phase 1 and number of blocks main memory can hold are shown below:

As we are holding data in bytes throughout the program, calculation is also done with the same measure
- Size of Input data T = 120,000,000 bytes.
- Memory size= 5MB (50% of given memory size for datasets and remaining 50% for program execution. i,e 2.5MB each).
- Therefore, below calculation is showed for 2.5MB

- For 5MB available memory size is 2.5MB,
- No of Blocks main memory can hold

= (memory size / block size)
= (2.5MB / 4KB)
= 625 Blocks.

- No of Tuples in T

=10,000,000

- No of Blocks in T

= (Total no. of tuples/No. of tuples in 1 Block)
= (10,000,000/341.34)
= 29,297 Blocks

- 30% of the memory Blocks will be used for Relation T

= (30% of No. of Block in Main Memory)
= (30*625)/100
= 188

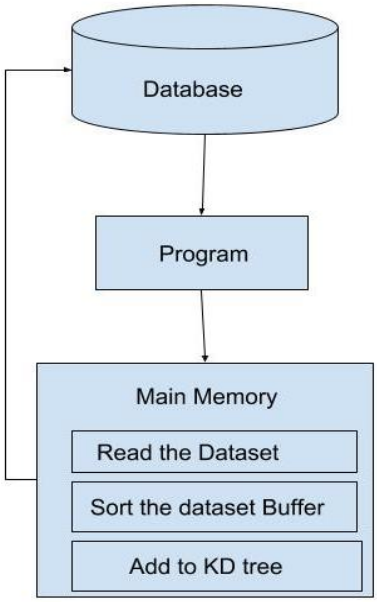
- Total number of blocks required for T

For T1= (T size/block size) = (120,000,000/4K) = 30000 Blocks.

The program calculates the following which are displayed in console:

- Main Memory Size.
- Total DISK I/O for both queries.
- Total Time taken for sorting of and building the KD tree and for traversing.
- Total Time taken for Each of the Queries.
- Total Execution Time for complete process.

Architecture Diagram



Algorithms

Here, we are allocating 50% of memory space for data and remaining 50% for program execution. So now consider we have 50% main memory available.

Algorithm:

1. Start.
2. In the level 0, sort the dataset with ‘X’
3. Take the mean of the resulted set, place it as the root
4. In the next level, sort the left dataset and right dataset with ‘Y’
5. Find the mean for both of the datasets and place them at the root
6. Similarly, for the next level, sort all 4 datasets with ‘Z’
7. Find the means for all the datasets and place at the root for all datasets
8. Cycle this thus making the KD tree with all datasets added as children's

   For Query1:

9. Read the dataset and form the KD tree with the above steps
10. Traverse through the tree for the given ranges
11. In the level 0, check the range with respect to ‘X’ and follow the roots
12. In the next level check with ‘Y’ and follow the roots
13. Finally do the same for the next level with ‘Z’
14. Repeat steps 11,12, and 13 for the end range point and thus find the region
15. In this, Traverse the tree finding all the points between the start point and the end point

    For query2

16. Read the dataset and form the KD tree by following steps 1 through 8
17. Take the Point for which to find the nearest neighboring point
18. Traverse through the KD tree Finding the point
19. Follow through the nodes after the point
20. Calculate the SSD for all the points after the given point to which to find the nearest neighbor
21. Take the point which has the least SSD and that is the nearest neighboring point for the given point
22. Calculate number of joined tuples and total number of Disk I/O.
23. End.

Coding standards Used

The most general coding conventions have been followed while developing the codes which are as follows,

- The name of the classes starts with a upper case character.

E.g.: Runner.java

- Constants are named with upper case characters and include underscore between two words (if applicable).
- The name of the variables is descriptive and are written in lower case including a capital letter to separate between words.
- The name of the methods starts with a lower-case character and use uppercase letters to separate words.

1. **Description of classes**

- **Runner.java:**

  This is the Main class, Reads the input dataset and makes the KD tree

- **KDTree.java:**

  Reads the input data and initializes the KD tree

- **Node.java:**

  Consists of Tree node with root, left node and right node and this is used to build the Tree

- **Point3D.java**

  This consists of points (X, Y,Z) and these values are then used to sort the tree at different levels with different coordinates

Results

Reading Data
Building KD Tree
Size of KD Tree: 10000001

Range Searching between for (357.3839, 512.5141, 626.4071) <==> (857.3839, 612.5141, 638.4071)
Found 6071 in range
Do you want to see them (Y/N): N

Looking for nearest neighbor of (0.993822, 4.021896, 2.58316)
(1.003822, 4.121896, 2.78316)