Bangladesh University of Engineering & Technology
Department of Computer Science & Engineering
CSE 464: Computational Geometry Sessional
Session January 2018

Assignment 4
2-dimensional KD tree

May 23, 2018

1 Introduction

In this assignment you will implement a geometric data structure called 2D tree (2-dimensional KD tree). You need to perform the following tasks.

1. Construct 2D tree from a set of points.
2. Perform rectangular range query to report the number of points and all the points inside the axis aligned rectangle.
3. Find nearest neighbor of a query point.

You can find the algorithms for task 1 and 2 in chapter 5 of [1]. The algorithm to solve task 3 is given in the slide provided.

2 Input

You will have to take input from a file. The first line of the input is a number n indicating the number of points. Each of the next n lines contains a pair of numbers indicating the x-coordinate and y-coordinate of each point. **No two points will have the same x-coordinate and/or y-coordinate**. The next few lines will consist of some queries. Each of query line will start with a character which can be **R** or **N** indicating Rectangular Range Query and Nearest Neighbor Query respectively. For Rectangular Range Query, four numbers will follow the character 'R' in the same line. The first two number indicate the leftmost bottom corner and the last two number indicate rightmost top corner of the rectangle. For Nearest Neighborhood Query, two number will follow after 'N', representing the x and y coordinate of the query point.

3 Output

You will have to show output both graphically and in console.

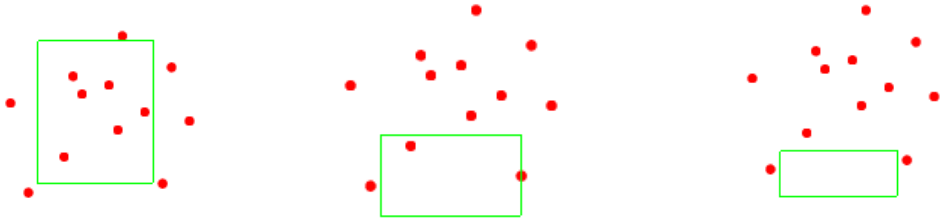
- **Console:** For Rectangular Range Query, you will need to print the co-ordinates of the points found in the query region. In the following line print the number of points inside the query region. For Nearest Neighborhood Query, you will have to print the co-ordinates of the nearest neighbor (in case of multiple nearest neighbor, only one will suffice).
- **Graphically:** For Rectangular Range Query you have to show the points and the rectangle graphically. For Nearest Neighbor Query, you have to show the query point and a circle centered at query point with radius equal to the distance from the query point to its nearest neighbor. You will process and display the result of a query after a button is pressed.

4 Sample IO:

The Table 1 contains one sample input output. The Figure 1 and 2 demonstrates sample graphical output for Range Queries and Nearest Neighbor Quereis respectively.

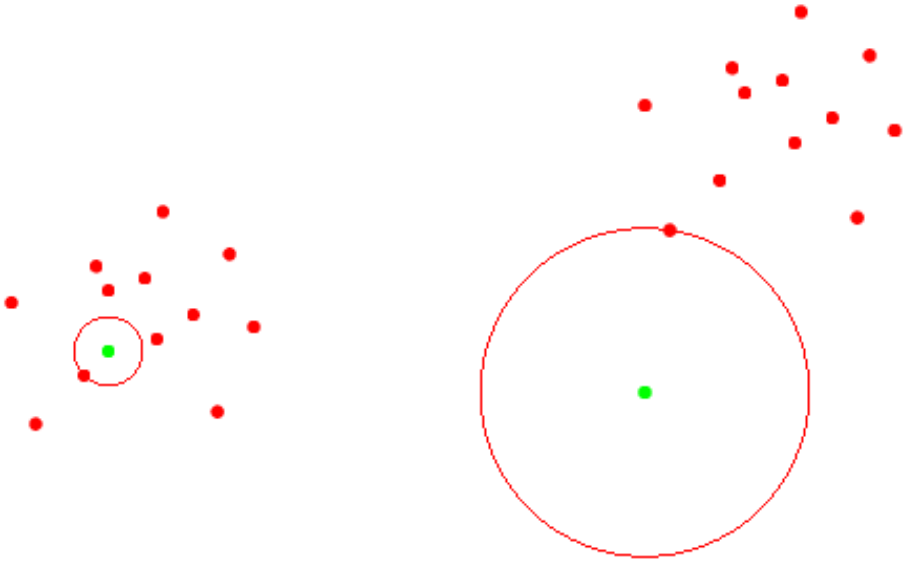
Input	**Output**
12 -2 4 1 5 -4 -3 10 1 5 2 -10 3 7 -6 2 0 2.5 10.5 -3 6 8 7 -8 -7 R -7 6 -6 10 R -7 7 -10 -2 R -7 6 -10 -5 N -2 -1 N -10 -20	(-4 , -3) (-3 , 6) (-2 , 4) (1 , 5) (2 , 0) (5 , 2) 6 (-4 , -3) (7 , -6) 2 0 2.82843 (-4 , -3) 13.1529 (-8 , -7)

Table 1: Sample Input Output



(a) Range Query 1 (b) Range Query 2 (c) Range Query 3

Figure 1: Graphical output for Range Queries in sample input.



(a) NN Query 1 (b) NN Query 2

Figure 2: Graphical output for NN Queries in sample input.

5 Important Notes

Please follow the instructions listed below while implementing your assignment:

- Implement using C++/Java programming language.
- Be cautious about floating point arithmetic.

6 Marks Distribution

- Constructing 2D tree: 6
- Rectangular Range Query: 6
- Nearest Neighbor Query: 5
- Graphical Output: 3
- Bonus: Handling points with same x and/or y coordinate

7 Rules

- You have to submit all your source codes via moodle. All the file name will be in following format

  <your 7 digit student id>_<additional name>

  For example, the submitted file name would look like 1305999_2Dtree.cpp if it is submitted by a student having 1305999 as student id. Put all your source files in a folder (even if you put all your codes in only one file) named after your 7 digit student id and create a **zipped** archive of the folder. Then submit the zip file in moodle. Failure to submit properly will cause 10% deduction of marks.

- Any type of plagiarism is strongly forbidden. -100% marks will be given to the students who will be found to be involved in plagiarism (from book, internet, from senior/classmates code etc.). It does not matter who is the server and who is the client.

8 Deadline

Deadline is set at 10:00 pm, July 6, 2018 for all lab groups.

References

[1] M. d. Berg, O. Cheong, M. v. Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.