

CSL 201, IIT Delhi
Assignment 4 - Part I
Searching Points in a one dimensional(1-D) range

Announce Date : 04.04.2014

Due Date : 12.04.2014

This part of the assignment is based on one dimensional(1-D) range search.

1 Problem Intuition

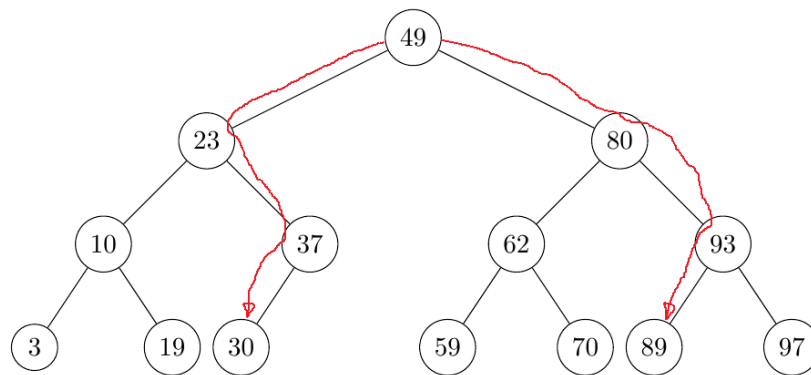
1-D range searches are used to answer queries like :

Given a set of points on x axis (1-D), and given a *closed* interval on x axis, output the points lying in the interval.

The data structure used is a balanced binary search tree (BBST) where the leaves and the internal nodes represent the actual points on the x axis. It is important to note that, the structure is independent of the query interval.

1.1 Example

Given a set of points $P = \{62, 49, 10, 97, 37, 23, 3, 89, 70, 19, 93, 59, 30, 80\}$, BBST would be:



And a query can be : Output all the points contained in an interval $[25, 90]$ (both ends are inclusive).

The output should contain this set of points : $\{30, 37, 49, 59, 62, 70, 80, 89\}$

Please note that the BBST shown above is just an example and any particular implementation like AVL tree for the above example may look different.

2 Assignment

Implement a program to

1. Build a BBST
2. Perform search operations for a given interval

2.1 Input format details

- You must use only cin/scanf.
- You must NOT use any file operations like fopen / fscanf etc. Input will be redirected from a file at the time of running of the program.
- The coordinates can be any (positive/negative) integer.
- Multiple points having same coordinates will be treated as a single point for the whole assignment.

2.1.1 Format

1. First line contains an integer 'N' - denoting number of points on x axis.
2. Next 'N' lines contain 'N' integers - one number per line - representing x coordinates of the points.
3. Next line contains lower x-coordinate of the interval.
4. Next line contains upper x-coordinate of the interval.

2.2 Output format details

You must use only cout/printf. You must NOT write output to any file.

2.2.1 Format

- Each line should contain x-coordinate of the point - 1 point per line.
- Points should be in the increasing order of their x-coordinates.

3 Example Input/Output

For the example explained above, input would look like :

```
14
62
49
10
97
37
23
3
89
70
19
93
59
30
80
25
90
```

and output would look like :

```
30
37
49
59
62
70
80
89
```

- You must use Linux to develop the assignment.
- You must prepare a modular makefile - it should have separate targets for compiling, linking and running. Targets will have dependencies among themselves. It should also contain a 'clean' target.
- Submit 'your' source file with the name ENTRY_NUMBER.cpp, a makefile and a documentation with the name ENTRY_NUMBER.txt. Along with other things, the documentation should include a detailed analysis of the complexity of your procedure in terms of the total number of points and the number points in the output. Don't submit any file that is provided to you - like I/O modules.

4 Notes

- The last part of this assignment will ask you to extend your search procedure to search for points in two or higher dimensional space. Hence you may implement your program in such a way that you can extend this later to more than 1 dimension.
- You may find it useful to use “standard template library” (STL) from C++.

STL reference can be found at : www.cplusplus.com/reference/stl

You can use any container from STL as long as you do not directly use any container to implement BBST.

You can use `std::sort` function.

- For input and output, you are ONLY allowed to use the IO modules provided to you. The functions in the IO module should not be added to your main program file. They need to be linked through the header file provided.