

## Data Structures 2014 Coursework

### Topic

The purpose of this coursework is to implement a quadtree that enables insertion; finding and deletion of integer valued two-dimensional points. Quadtree is a data structure that can be considered as an extension to a binary tree. In a binary tree the objects to be inserted use a single valued key while quadtrees use two-dimensional keys. Programming language used is Java.

### Quadtree

Quadtree and its three-dimensional counterpart octree are widely used in different geometric applications. For instance, in computer graphics the computation of visibility of objects and their potential collisions can be speeded up with quadtrees. The basic idea of a quadtree is presented in figures 1 and 2. The whole area of figure 1 presents the entire area to be subdivided. The first level of subdivision (blue) divides the area into four equally sized quadrants 1, 2, 3 and 4. Furthermore, the second quadrant of blue area is divided into four areas (green). Finally the first quadrant of green area is divided into four areas (orange). The black dots in figure 1 are points that reside in respective places in the original area. Places of the dots in a quadtree are presented in figure 2.

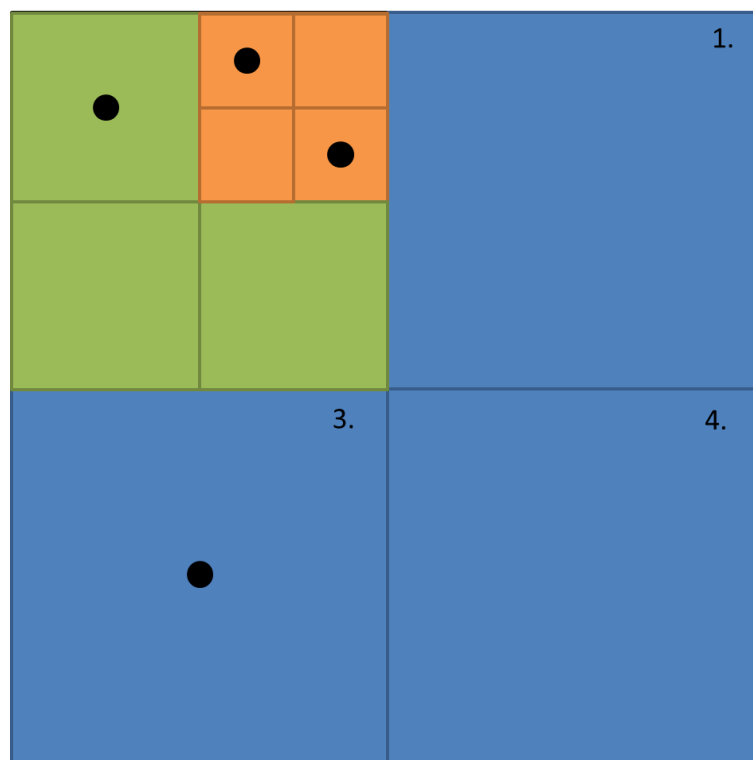


Figure 1 Successive division of original area into four equally size areas

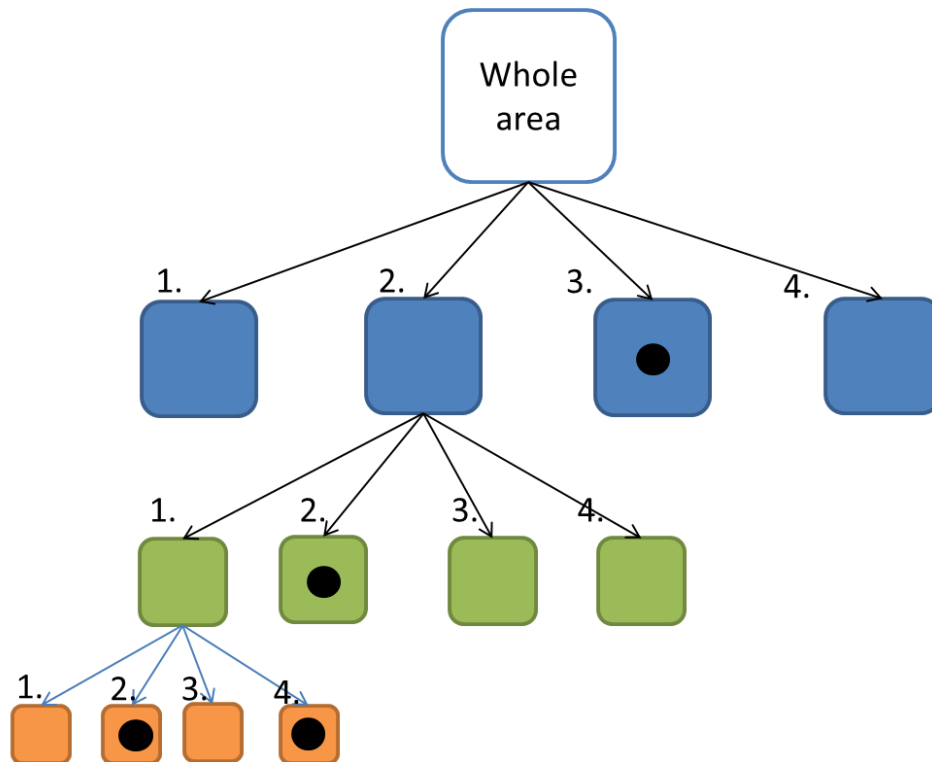


Figure 2 A quadtree representation of successive divisions of Fig 1

### Implementation

In this work we use classes Tira.java, QuadTree.java and Node.java. It is not allowed to use existing classes - they can only be used for reading files and outputting text to the screen. The original size of area is some power of two. In this case the subdivision is easy. The default size for original area is 1024x1024 and it is given in the constructor of QuadTree.java. Use Tira.java and testdata.txt files as a starting point to your work.

Class Tira.java reads ten integer valued points from testdata.txt file. These points can be inserted to the tree, they can be searched for and they can be removed from the tree. Class QuadTree.java takes care of inserting (Insert(p)), finding (Find(p)) and removing (Remove(p)). If user is trying to remove point that is not in the tree he/she will be informed. If we can find a point, the upper left and lower right corners of area that contain the point will be displayed in the command prompt on a single line. After each insertion and deletion the content of the tree will be displayed to the user in inorder. The origin of area to be divided is in its upper left corner. The values of x-coordinates increase from left to right and the values of y-coordinate increase from top to bottom. Class Node.java contains the necessary pointers to child nodes and methods to set the pointers. In addition, the node has a storage for a single point and the Node can tell whether a point to be inserted belong to the area of node or not. Material about the quadtrees can be found for instance from <http://www.cs.umd.edu/~hjs/pubs/Samet85k.pdf>. Look also <http://www.youtube.com/watch?v=fuexOsLOfI0>.

### Returning the work

Final work should be returned before 31.1.2015 to the address jyrki.rasku(at)uta.fi. The topic of the returning message is Tira2014. Modify your program such that the commented lines in Tira.java are executable. Uncomment the lines such that the program can be compiled and run with testdata.txt file. Please pack all your files into a single file using 7z.

### **About grading**

Work that arrives in time and does the job mentioned above is worth ten points. Errors in Insert, Find and Remove methods reduce amount of points.