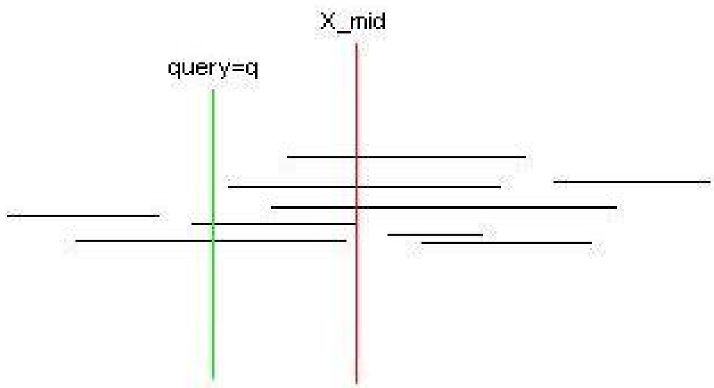
Interval Trees

Scribed by: Atul Bansal

6 Oct,2006

1 One Dimensional Case

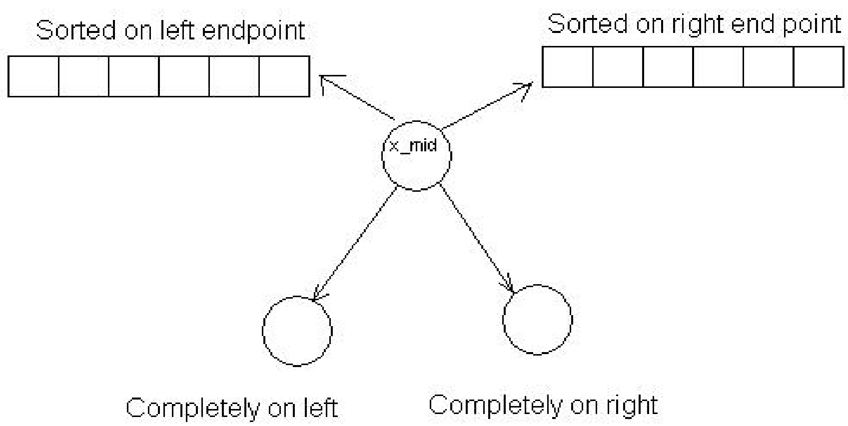


1 Dimensional Case

1.1 Problem Definition

The input is a set of n intervals along x axis and we are given an infinite verticle line and a particular value x as shown above.The output is the set of intervals which intersect this verticle line.

1.2 Algorithm and Data Structure



The algorithm is as follows:

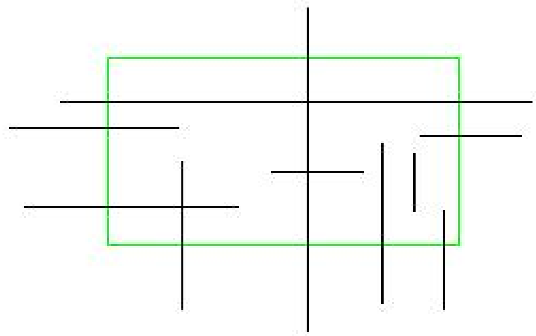
1. If $q \leq x_{mid}$ then output those points on the left sorted array which have their left end points to the left of q . Then recurse on the left subtree.

2. Else output those points on the right sorted array which have their right end points to the right of q . Then recurse on the right subtree.

1.3 Analysis

Since arrays stored at node for x_{mid} the time spent at a node is $\leq$ (1 + No. of Intervals outputted). Hence running time is $O(k + log(n))$, where k is the number of Intervals outputted. Also we have: Space: $O(n)$ Preprocessing time: $O(nlog(n))$

2 Two Dimensional Case

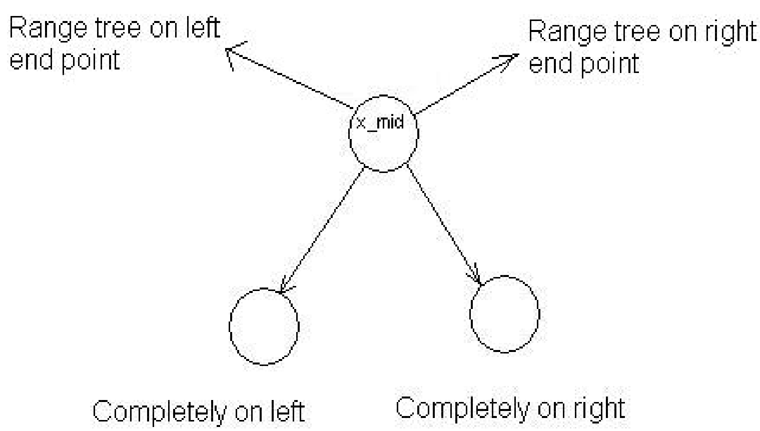


2 Dimensional case

2.1 Problem Definition

The input is a set of n intervals which are either verticle or horizontal. A query gives a rectangle , with sides parallel to the axes and asks for the intervals which intersect this rectangle.

2.2 Algorithm and Data Structure

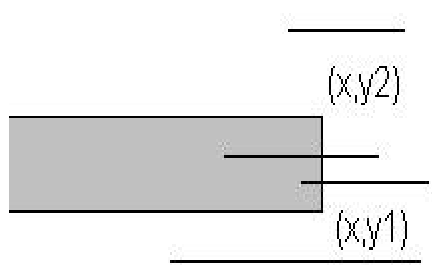


The algorithm is as follows:

1. If $q \leq x_{mid}$ then perform range query on the range tree for left end points as shown. Then recurse on the left subtree.

2. Else perform range query on the range tree for right end points as shown. Then recurse on the right subtree.

The range query is performed for the infinite rectangle as shown below:



Intervals with left end point in rectangle are required

2.3 Analysis

The height of range tree and interval tree is $O(log(n))$. Hence running time for a query is $O(log^2(n) + k)$ where k is the size of output. Space requirement is $O(nlog(n))$.