Precept Topics

- Hash Tables
- K-d Trees

Relevant Material

- Book chapters: 3.4.

A. RECAP: Hash Tables and k-d Trees

Your preceptor will review the material covered in this week's lectures. Feel free to use this space for notes.

B. EXERCISES: Hash Tables (~15 minutes)

Solve the Exercises 1 to 6 in the  Hash Tables Ed lesson.

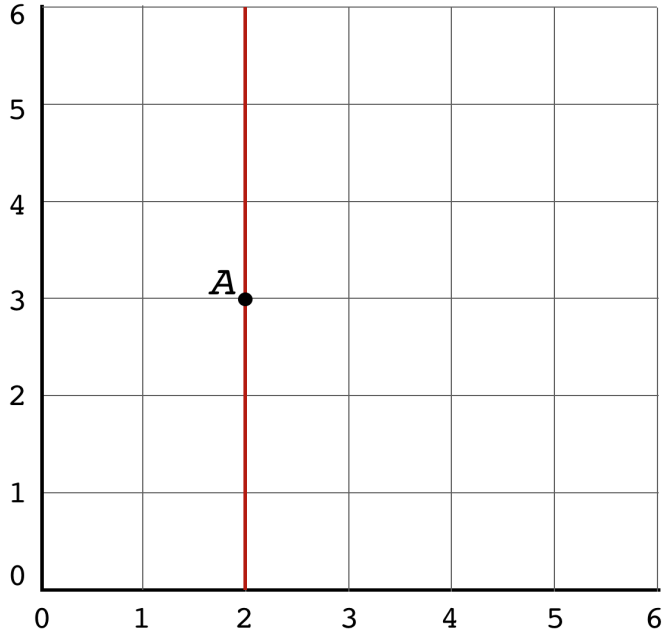
C. EXERCISE: Kd-Trees (~40 minutes)

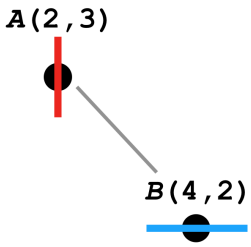
(a) Draw the Kd-tree that results from inserting the following points:

A	B	C	D	E	F	G
(2, 3)	(4, 2)	(4, 5)	(3, 3)	(1, 5)	(4, 4)	(1, 1)

Draw each point on the grid, as well as the vertical or horizontal line that runs through the point and partitions the plane or a subregion thereof.

Note: While inserting, go left if the coordinate of the inserted point is less than the coordinate of the current node. Go right if it is greater than ***or equal***.



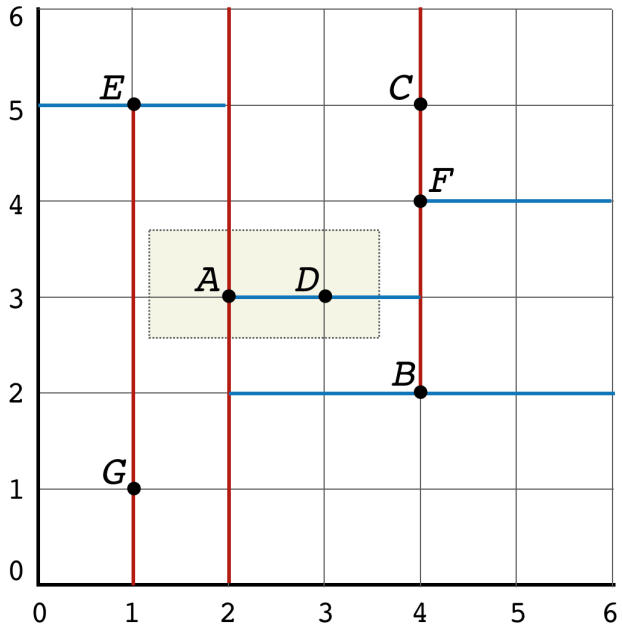


(b) Determine each point's bounding box.

A	$[-\infty, \infty] \times [-\infty, \infty]$
B	
C	
D	
E	
F	
G	

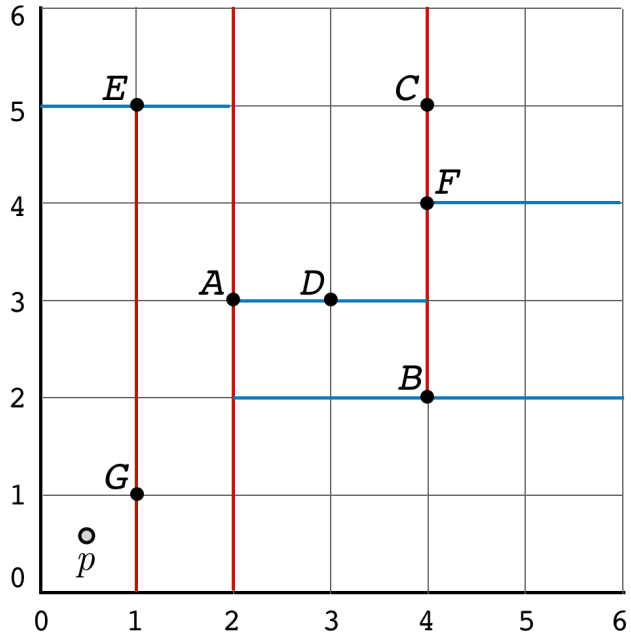
(c) Number the (non-null) nodes in the sequence they are visited by a *range query* with the rectangle shown below. Which subtrees are pruned? (Some null subtrees may be pruned, and some may not be.)

Remember. The range search algorithm recursively searches in both the left and right subtrees unless the bounding box of the *current* node does not intersect the query rectangle. If both do, our convention is to visit the left one first.



(d) Number the (non-null) nodes that in the order they are visited during a *nearest neighbor (NN) query* using the point p shown below. Which subtrees are pruned?

Remember. The NN algorithm recursively searches in both the left and right subtrees unless the distance between p and the bounding box of the *current* node is at least the distance between p and the nearest point found so far.



D. ASSIGNMENT OVERVIEW: Implementing 2-d Trees

If there is time left, your preceptor will overview the assignment due next Monday and give some implementation tips.