

Running Time of Every Data Structure So Far:

	Unsorted Array	Sorted Array	Unsorted List	Sorted List
Find				
Insert				
Remove				
Traverse				

	Binary Tree	BST	AVL
Find			
Insert			
Remove			
Traverse			

Iterators and MP4

Three weeks ago, you saw that you can use an iterator to loop through data:

```

1 DFS dfs(...);
2 for ( ImageTraversal::Iterator it = dfs.begin();
      it != dfs.end(); ++it ) {
3     std::cout << (*it) << std::endl;
4 }

```

You will use iterators extensively in MP4, creating them in Part 1 and then utilizing them in Part 2. Given the iterator, you can use the for-each syntax available to you in C++:

```

1 DFS dfs(...);
2 for ( const Point & p : dfs ) {
3     std::cout << p << std::endl;
4 }

```

The exact code you might use will have a generic `ImageTraversal`:

```

1 ImageTraversal & traversal = /* ... */;
2 for ( const Point & p : traversal ) {
3     std::cout << p << std::endl;
4 }

```

Looking forward to Final Project

Q: What is a graph and why do we care?

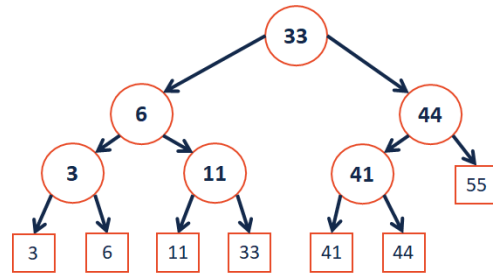
Start finding a team and brainstorming ideas now.

Range-based Searches:

Q: Consider points in 1D: $p = \{p_1, p_2, \dots, p_n\}$.
...what points fall in $[11, 42]$?



Tree Construction:

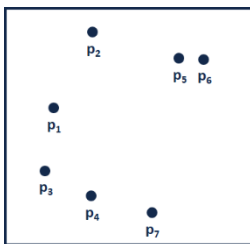


Range-based Searches:

Running Time:

Extending to k-dimensions:

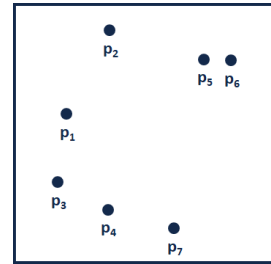
Consider points in 2D: $\mathbf{p} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$:



...what points are inside a range (rectangle)?
...what is the nearest point to a query point \mathbf{q} ?

kd-Tree Motivation:

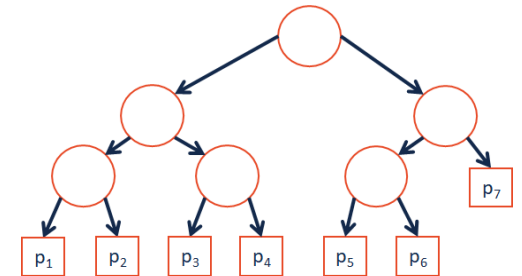
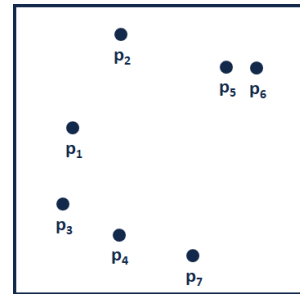
First, let's try and divide our space up:



kd-Tree Construction:

How many dimensions exist in our input space?

How do we want to "order" our dimensions?



CS 225 – Things To Be Doing:

1. mp_traversal
2. Find teammates and data sets for the final project
3. lab_avl released tomorrow
4. Daily POTDs are ongoing!