Range-based Searches:
Q: Consider points in 1D: \( p = \{ p_1, p_2, \ldots, p_n \} \).
...what points fall in \([11, 42]\)?

Tree Construction:

kd-Tree Motivation:
First, let’s try and divide our space up:

Range-based Searches:

Tree Construction:

Running Time:

Extending to \( k \)-dimensions:
Consider points in 2D: \( p = \{ p_1, p_2, \ldots, p_n \} \):
...what points are inside a range (rectangle)?
...what is the nearest point to a query point \( q \)?
Motivation
Can we always fit our data in main memory?

Where else do we keep our data?

vs. CPU: 3 GHz == 3m ops / __________. * ___ cores

AVL Operations on Disk:

How deep do AVL trees get?

BTree Motivations
Knowing that we have long seek times for data, we want to build a data structure with two (related) properties:

1.

2.

BTree_m

Goal: Build a tree that uses ________________ /node!

...optimize the algorithm for your platform!

A BTree of order m is an m-way tree where:
1. All keys within a node are ordered.
2. All leaves contain no more than m-1 nodes.

BTree Insert, using m=5

...when a BTree node reaches m keys:

<table>
<thead>
<tr>
<th>CS 225 – Things To Be Doing:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Programming Exam B starts next Tuesday (March 13th)</td>
</tr>
<tr>
<td>2. MP4 extra credit ongoing (final deadline March 12th)</td>
</tr>
<tr>
<td>3. lab_avl released this week; lab sections are being held this week!</td>
</tr>
<tr>
<td>4. Daily POTDs are ongoing!</td>
</tr>
</tbody>
</table>