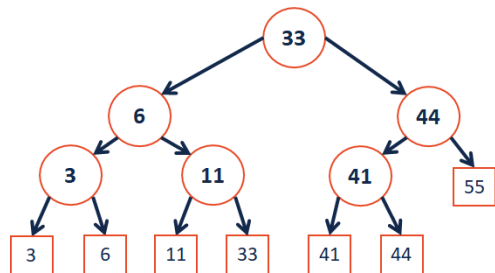**#17: kd-Trees**

Range-based Searches:

Q: Consider points in 1D: p = {p_1, p_2, ..., p_n}.

...what points fall in [11, 42]?
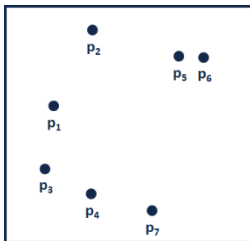

Tree Construction:

Range-based Searches:

Running Time:

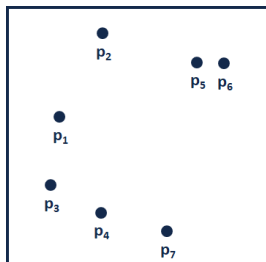## Extending to k-dimensions:

Consider points in 2D: **p = {p_1, p_2, ..., p_n}:**

...what points are inside a range (rectangle)?
...what is the nearest point to a query point **q**?
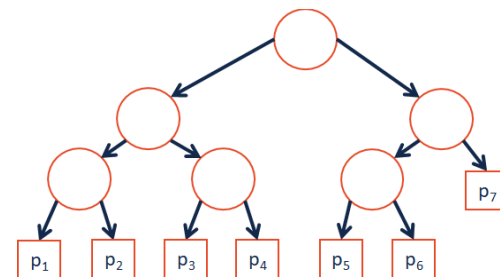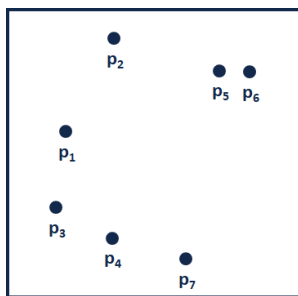
kd-Tree Motivation:

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?

Motivation

Can we always fit our data in main memory?

Where else do we keep our data?