CS 225

#24: BTree Analysis

October 18, 2021 · *G Carl Evans*

BTree Properties

For a BTree of order **m**:

1. All keys within a node are ordered.
2. All leaves contain no more than **m-1** nodes.
3. All internal nodes have exactly **one more children than keys**.
4. Root nodes can be a leaf or have **[2, m]** children.
5. All non-root, internal nodes have **[ceil(m/2), m]** children.
6. All leaves are on the same level.

BTree Analysis

The height of the BTree determines maximum number of ____________ possible in search data.

...and the height of our structure:

Therefore, the number of seeks is no more than: ___________.

...suppose we want to prove this!

BTree Proof #1

In our AVL Analysis, we saw finding an **upper bound** on the height (**h** given **n,** aka **h = f(n)**) is the same as finding a **lower bound** on the keys (**n** given **h,** aka $\mathbf{f^{-1}(h)}$).

Goal: We want to find a relationship for BTrees between the number of keys (**n**) and the height (**h**).

BTree Strategy:

1. Define a function that counts the minimum number of nodes in a BTree of a given order.
   a. Account for the minimum number of keys per node.
2. Proving a minimum number of nodes provides us with an upper-bound for the maximum possible height.

Proof:

1a. The minimum number of <u>nodes</u> for a BTree of order **m** at each level is as follows:

root:

level 1:

level 2:

level 3:

...

level h:

1b. The minimum total number of <u>nodes</u> is the sum of all levels:

---

2. The minimum number of keys:

---

3. Finally, we show an upper-bound on height:

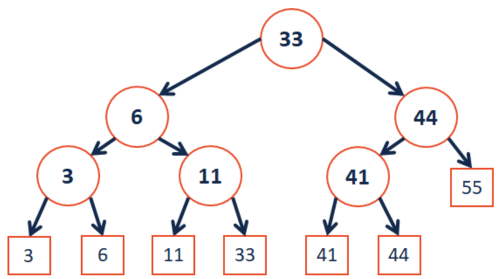
So, how good are BTrees?
Given a BTree of order 101, how much can we store in a tree of height=4?

Minimum:

Maximum:

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

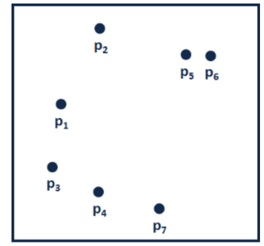
Tree Construction:



Range-based Searches:

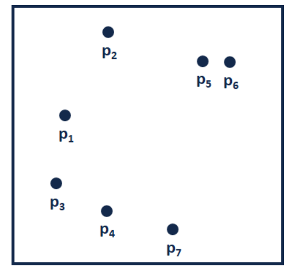
Running Time:

Extending to k-dimensions:
Consider points in 2D: $\mathbf{p = \{p_1, p_2, \ldots, p_n\}}$**:**



...what points are inside a range (rectangle)?
...what is the nearest point to a query point **q**?

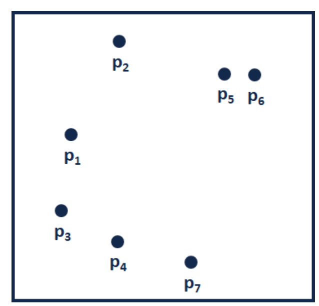
kd-Tree Motivation:
First, let's try and divide our space up:

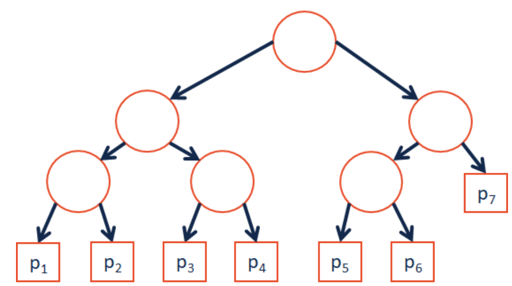


---

kd-Tree Construction:
How many dimensions exist in our input space?

How do we want to "order" our dimensions?





CS 225 – Things To Be Doing:
1. Mp_traversals due today **2.** Potds ongoing **3.** Exam 2 practice releases on Tuesday