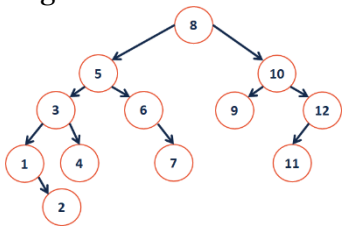
CS 225

# #19: BTrees

## BTree Motivation

Big-O assumes uniform time for all operations, but this isn't always true.

However, seeking data from the cloud may take 100ms+.
...an O(lg(n)) AVL tree no longer looks great:




---

## BTree Motivations

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

1.

2.

3.

**BTree$_m$**

| -3 | 8 | 23 | 25 | 31 | 42 | 43 | 55 |
|---|---|---|---|---|---|---|---|

m=9

**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.** Nodes contain up to _____ keys and have _____ children

**3.** All leaves in a BTree are on the same level

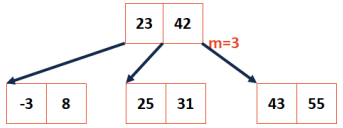
---

## BTree Insert, using m=5

...when a BTree node reaches **m** keys:

---

## BTree Insert, m=3:




---

## Great interactive visualization of BTrees:

https://www.cs.usfca.edu/~galles/visualization/BTree.html

**BTree Properties**

For a BTree of order **m**, there are additional bounds on the size of nodes:

1. Root nodes can be a leaf or have _____ children.
2. All non-root, internal nodes have _____ children.