**Motivation**

Can we always fit our data in main memory?

Where else do we keep our data?

---

**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**

---

**AVL Operations on Disk:**

How deep do AVL trees get?

---

**Example #2:**
**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 key than children**.
4. Root nodes can be a leaf or have $[2, m]$ children.
5. All non-root, internal nodes have $[\lceil m/2 \rceil, m]$ children.
6. All leaves are on the same level.

---

**BTree Search**

```
bool Btree::_exists(BTreeNode & node, const K & key) {
    unsigned i;
    for (i=0; i<node.keys_ct_ && key<node.keys_[i]; i++) { }
    if ( i < node.keys_ct_ && key == node.keys_[i] ) {
        return true;
    }
    if ( node.isLeaf() ) {
        return false;
    } else {
        BTreeNode nextChild = node._fetchChild(i);
        return _exists(nextChild, key);
    }
}
```

---

**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 Setup**

In our AVL Analysis, we saw finding an upper bound on the height (given $n$) is the same as finding a lower bound on the nodes (given $h$).

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

---

**CS 225 – Things To Be Doing:**

1. Exam #7 is live!
2. MP4 due tonight
3. New lab coming Wednesday!
4. Daily POTDs