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:

Consider Instagram profile data:

<table>
<thead>
<tr>
<th>How many profiles?</th>
<th>How much data /profile?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVL Tree</td>
<td>BTree</td>
</tr>
<tr>
<td>Tree Height</td>
<td></td>
</tr>
</tbody>
</table>

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 Insert, using m=5

...when a BTree node reaches m keys:

Great interactive visualization of BTrees:
[https://www.cs.usfca.edu/~galles/visualization/BTree.html](https://www.cs.usfca.edu/~galles/visualization/BTree.html)
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.

Example BTree

What properties do we know about this BTree?

---

BTree Search

---

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 Analysis
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. Programming Exam 2 reservations open in CBTF
2. Mp_traversal due next Monday (Oct. 18)
3. lab_avl due Sunday
4. Daily POTDs are ongoing!