CS 225
Data Structures

Oct. 25 – BTree Analysis
BTree
Btree Properties

A **BTrees** of order $m$ is an $m$-way tree:
- All keys within a node are ordered
- All nodes contain no more than $m-1$ keys.

- All internal nodes have exactly **one more child than keys**
- Root nodes can be a leaf or have $[2, m]$ children.
- All non-root, internal nodes have $[\lceil m/2 \rceil, m]$ children.

- 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 the structure is: ________________.

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 \( h \)).

We want to find a relationship for BTrees between the number of keys (\( n \)) and the height (\( h \)).
BTree Analysis

**Strategy:**
We will first count the number of nodes, level by level.

Then, we will add the minimum number of keys per node ($n$).

The minimum number of nodes will tell us the largest possible height ($h$), allowing us to find an upper-bound on height.
BTree Analysis

The minimum number of nodes for a BTree of order m at each level:

root:

level 1:

level 2:

level 3:

...

level h:
BTree Analysis

The total number of nodes is the sum of all of the levels:
BTree Analysis

The total number of keys:
BTree Analysis

The **smallest total number of keys** is:

So an inequality about $n$, the total number of keys:

Solving for $h$, since $h$ is the number of seek operations:
BTree Analysis

Given $m=101$, a tree of height $h=4$ has:

Minimum Keys:

Maximum Keys:
<table>
<thead>
<tr>
<th>Locker Number</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>103</td>
<td>Rick</td>
</tr>
<tr>
<td>92</td>
<td>Kiri</td>
</tr>
<tr>
<td>330</td>
<td>Mary Catherine</td>
</tr>
<tr>
<td>46</td>
<td>Blake</td>
</tr>
<tr>
<td>124</td>
<td>Erin</td>
</tr>
</tbody>
</table>
Hashing

Commonly:
Just use a “hash table” to implement a _______________. 
Dictionary ADT

Data is often organized into key/value pairs:

UIN ➔ Advising Record
Course Number ➔ Lecture/Lab Schedule
Node ➔ Incident Edges
Flight Number ➔ Arrival Information
URL ➔ HTML Page
...

...
#ifndef DICTIONARY_H
#define DICTIONARY_H

template <class K, class V>
class Dictionary {
public:
    void insert(K & k, V & v);
    void remove(const K & k);
    V & find(const K & k) const;

private:
};
#endif
Hashing

Goals:
We want to define a *keyspace*, a (mathematical) description of the keys for a set of data.

...use a function to map the *keyspace* into a small set of integers.
Exam 7 is ongoing!
More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

MP5: Available now!
Extra Credit +7 deadline: Monday, Oct. 30

Lab: lab_btree
Due Sunday, Oct. 29 at 11:59pm

POTD
Every Monday-Friday – Worth +1 Extra Credit /problem (up to +40 total)