Chapter 5, Binary search trees:

- Binary search trees; the balanced BST problem (Oct 14 & 21)
- AVL trees (Oct 21 & 24)
- Traditional red-black trees (last week Wed, Oct 26)
- Left-leaning red-black trees (last week Fri, Oct 28)
- “Wrap-up” BSTs, B-trees (Today)
- Begin dynamic programming (Wednesday)
- Test 2 Friday, Nov 11

Today:

- Finish red-black trees
- B-tree origin stories
  - Two-three trees
  - Sorted arrays
  - Linked/array hybrid
- B-tree definition
- B-tree implementation
Subtree with keys less than ANT
Subtree with keys between ANT and BEE
Subtree with keys greater than WASP
Formally, a B-tree with maximum degree $M$ over some ordered key type is either

- empty, or
- a node with $d - 1$ keys and $d$ children, designated as lists `keys` and `children` such that
  - $\lceil M/2 \rceil \leq d \leq M$,
  - `children[0]` is a B-tree such that all of the keys in that tree are less than `keys[0]`,
  - for all $i \in [1, d - 1)$, `children[i]` is a B-tree such that all of the keys in that tree are greater than `keys[i - 1]` and less than `keys[i]`,
  - and `children[d - 1]` is a B-tree such that all of the keys in that tree are greater than `keys[d - 2]`. 
keys
eight bytes each position
values
four bytes each position
children
eight bytes each position
deg

offset in struct
0 8 88 96 100 140 144 152 240 248 252

keys
ANT
BEE
WASP
values
96 100 140 1440 8 88 152 240 248 252

keys
ANT
BEE
WASP
values
5 12

keys
ANT
BEE
WASP
values
33

nodes
node for keys
less than
ANT
node for keys
between
ANT
and
BEE
node for keys
greater than
WASP

deg
13

offset 
in struct
node for keys
less than
ANT
node for keys
between
ANT
and
BEE
node for keys
greater than
WASP
\[(M - 1) \sum_{i=0}^{h-1} M^i = (M - 1) \frac{M^h - 1}{M - 1} = M^h - 1\]

keys per node

sum of nodes at each level

\[n = M^h - 1\]

\[M^h = n + 1\]

\[h = \log_M(n + 1)\]
\[ n = M^h - 1 \]
\[ M^h = n + 1 \]
\[ h = \log_M(n + 1) \]
\[ h = \log_{\frac{M}{2}}(n + 1) = \frac{\log_M(n + 1)}{1 - \log_M 2} \]
Cost of a search:

\[ \lg M \cdot h = \lg M \cdot \frac{\log_M(n+1)}{1-\log_M 2} \]

\[ = \lg M \cdot \frac{\log(n+1)}{1-\frac{\log 2}{\log M}} \]

\[ = \frac{\log(n+1)}{1-\frac{1}{\log M}} \]

\[ = \frac{\log M}{\log M - 1} \log(n + 1) \]

Compare: 1.44 \lg n for AVL trees, 2 \lg n for RB trees.
Let $c_0$ be the cost of searching at a node (proportional to $\lg M$) and $c_1$ be the cost of reading a node from memory. The the cost of an entire search is

$$(c_0 + c_1) \frac{\log_M(n + 1)}{1 - \log_M 2}$$

Now, consolidate the constants by letting $d = \frac{c_0 + c_1}{1 - \log_M 2}$, and we have

$$d \log_M(n + 1)$$
Coming up:

Do **Traditional RB** project *(suggested by Fri, Nov 4)*
*(Recommended: Do **Left-leaning RB** project for your own practice)*

*Due Mon, Oct 31* *(today* *(end of day)* *(but hopefully you’ve spread it out)*
Read Sections 5.(4-6) *[some parts carefully, some parts skim, some parts optional—see Schoology]*
Do Exercise 5.14
Take quiz

*Due Thurs, Nov 3* *(end of day)*
Read Section 6.(1&2)
Do Exercises 6.(5–7)
Take quiz

*Due Fri, Nov 4* *(end of day)*
Read Section 6.3
Do Exercises 6.(16, 19, 23, 33)
Take quiz