CIS 451/551

week 7: B+ tree insertion
index buzzwords

• *index attribute(s)*: search fields for index
• *clustering index*: underlying table is sorted on its index attributes

• *sparse index*: does not contain all possible values of the index attributes appearing in the table
• *dense index*: not sparse, contains all values
terminology

• ISAM better static structure
• B-trees introduced as more easily modifiable
• B+ trees have additional leaf layer
  – useful for range queries, index joins
• “What, if anything, the B stands for has never been established.” (from wikipedia)
• often thought of as standing for “balanced”
text’s rules (Ramakrishnan-Gehrke)

• d is the **order** of the tree (fixed up front)
• **internal node** of order d structure:
  \[ <P_0, K_1, P_1, K_2, ..., P_{m-1}, K_m, P_m> \]
• **leaf node**:
  \[ <P_{prev},(K_1,P_1), (K_2, P_2), ..., (K_m,P_m), P_{next}> \]
• for \( d \leq m \leq 2d \)
• except root allows \( 1 \leq m \leq 2d \)
tree shape

- root
- internal nodes
- leaf nodes
- pointers to record blocks
right bias on splitting

overflowing **leaf** node split

(a b c d e)

becomes

(c)

promote c by inserting into parent

(b)

(d e)

overflowing **internal** node split

(a b c d e)

becomes

(c)

similarly c gets promoted

(b)

(d e)

left “bias” would have nodes (abc) and (de), which is a bit more common

promoted value remains in leaf node, does not remain in internal node
now the problem

insert into empty B+ tree with order d=2 the values:

20, 40, 37, 27, 25, 30, 15, 60, 42, 35, 5, 22, 50, 11, 32, 36
recall with order d=2

• leaf node can have 2-4 keys, with 2-4 child pointers
• internal node can have 2-4 keys with 3-5 child pointers
• the root is an exception: it can have a single key
• we are using the B+ tree “rules” from our text
• so on search, “ties” go to the right
• other texts will have slightly different rules so beware
starting to insert

after 20: (20)

after 37: (20, 37, 40)

after 40: (20, 40)

after 27: (20, 27, 37, 40)
The first split

After 25 overflow, so need to split:

(20, 25, 27, 37, 40)

Splits to

(27)

(20, 25)  (27, 37, 40)

Then inserting 30 and 15 is boring:

(27)

(15, 20, 25)  (27, 30, 37, 40)
after 60 overflow again:

then inserting 42, 35, 5 is simple:
inserting 22 causes a split:
50 also causes a split:

\[(20, 27, 37)\]  
\[(20, 22, 25)\]  
\[(27, 30, 35)\]  
\[(37, 40, 42, 50, 60)\]  
\[(5, 15)\]  

becoming
inserting 11 and 32 again simple

note that many of the insertions involve little work
now inserting 36 will cause two splits

32 is promoted into parent, but then the root node overflows,
continued on next page
• the root may have only one key (and two children)
• the leaf nodes are linked together (not shown here)