CSC 225

Algorithms and Data Structures: I

Rich Little
rlittle@uvic.ca
ECS 516

Properties

We want to show three things are true:

1. After inserting a key k into a 2-3 tree with keys $k_1, \ldots, k_n$ by the steps discussed, the resulting tree is a 2-3 tree containing keys $k_1, \ldots, k_n, k$.

2. A 2-3 tree with n keys has exactly $n + 1$ external nodes

3. The height of the 2-3 tree is $\Theta(\log n)$

- The result being 2-3 search and insertion are $\Theta(\log n)$

Reminder: Definition 2-3 tree

- A *2-3 tree* is a *perfectly balanced* 2-3 search tree, which is one where all leaves have the same distance from the root.

1. After inserting a key $\boldsymbol{k}$ into a 2-3 tree with keys $\boldsymbol{k_1, \dots, k_n}$ the resulting tree is a 2-3 tree containing keys $\boldsymbol{k_1, \dots, k_n, k}$

- Recall, when inserting a key, the search for key k returns a leaf

- Case 1. If the leaf is root, then the tree is empty and the leaf (root node) is replaced by a 2-node with key k

- Otherwise, the search terminates in a leaf with parent node v

- We distinguish the cases where v is a 2-node (Case 2) and where v is a 3-node (Case 3)

Proof

- To show: After inserting a key into a 2-3 tree the tree remains
  A. a 2-3 search tree
  B. the tree is perfectly balanced
- Note that the internal node the search terminates in is always a parent of leaves only.
- Case 1. Inserting into an empty tree
- Case 2. Search terminates in a 2-node
- Case 3. Search terminates in a 3-node
  - Case 3.1. Search terminates at root
  - Case 3.2. Parent: 2-node
  - Case 3.3. Parent: 3-node

Case 1. Inserting into an empty tree

- To show: After inserting a key into a 2-3 tree the tree remains

  A. a 2-3 search tree

  B. the tree is perfectly balanced

- After inserting a key into an empty key, the key consists of a single 2-node. Properties A and B are satisfied

Proof

- To show: After inserting a key into a 2-3 tree the tree remains

  A. a 2-3 search tree

  B. the tree is perfectly balanced

- Case 2. Search terminates in a 2-node

- The number of internal nodes does not change. The node where the key is inserted is added a third leaf, keeping the tree perfectly balanced.

- Inserting the new key into the 2-node will maintain the search tree property: The search determined the right subtree for the key to be inserted. Inserting the key to the left of the 2-node key if smaller and to the right if larger will complete the insertion maintaining the search tree property.

Proof

- To show: After inserting a key into a 2-3 tree the tree remains

  A. a 2-3 search tree

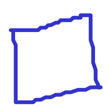
  B. the tree is perfectly balanced

- Case 3. Search terminates in a 3-node
  - Case 3.1. Search terminates at root
  - Case 3.2. Parent: 2-node
  - Case 3.3. Parent: 3-node

2. A 2-3 tree with n keys has exactly $n+1$ external nodes

let e = # of external nodes

Proof: By induction on the number of keys, n.

B.C.: $n=0$:  $e=1$, $n=0$, $e=n+1$.

$n=1$:  $e=2$, $n=1$, $e=n+1$

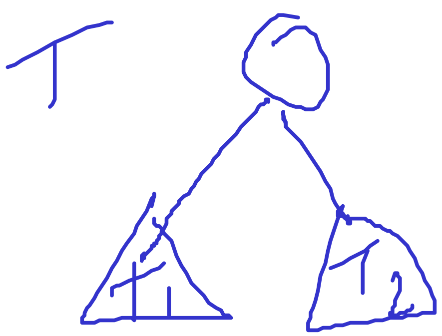
$n=2$:  $e=3$, $n=2$, $e=n+1$

I.H.: Assume $e=n+1$ $\forall$ trees with $n \le k$ keys.

2. Proof continued

IS: Let T be a 2-3 search tree with $n=k+1$ keys. Want show that $e=n+1$.

2case: Root is a 2-node. Remove root from T.



We have two 2-3 trees, T_1 & T_2 where T_1 has k_1 keys & T_2 has k_2 keys, s.t. $k_1+k_2=k$. Both $k_1, k_2 \le k$, so $e_1 = k_1+1$ and $e_2 = k_2+1$ by I.H. Number of ext nodes in T, $e=(k_1+1)+(k_2+1)$

$$e = (k_1+1)+(k_2+1) = k_1+k_2+1+1$$

$$= k+1+1 = n+$$

3. The height, h, of the 2-3 tree is $\Theta(\log n)$

Proof:

Let T be a 2-3 tree with n keys and height h.
T has $n+1$ exte. nodes all at depth h.

Min. 2-3 tree of height h has 2^h ext. nodes.
Max 2-3 tree of height h has 3^h ext. nodes

Thus, $2^h \le n+1 \le 3^h$

$2^h \le n+1$

$h \le \log_2(n+1)$ ∴ $h \in O(\log n)$

$n+1 \le 3^h$

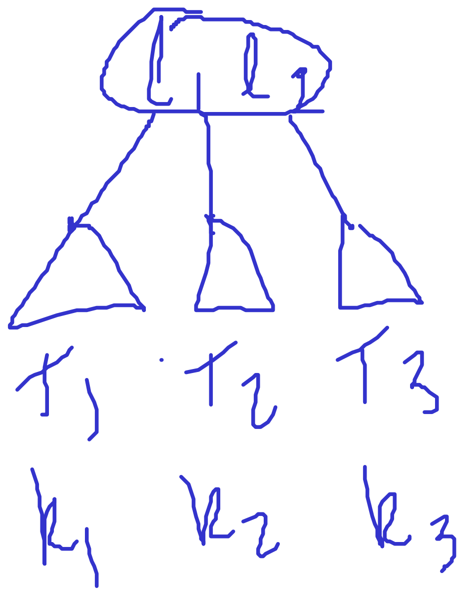
$\log_3(n+1) \le h$

$h \in \Omega(\log n)$

3. Proof continued

$n = k+1$ keys in T

Case 2: Root of T is a 3-node.



Remove root from T, leaving T_1, T_2, T_3 with total $k_1 + k_2 + k_3 = k-1$ keys.

So, $k_1, k_2, k_3 < k$ and ind. hyp. holds.

$\ell_1 = k_1 + 1$, $\ell_2 = k_2 + 1$, $\ell_3 = k_3 + 1$

In T,

$$\ell = (k_1+1) + (k_2+1) + (k_3+1)$$
$$= (k-1) + 3 = k+2 = n+1$$