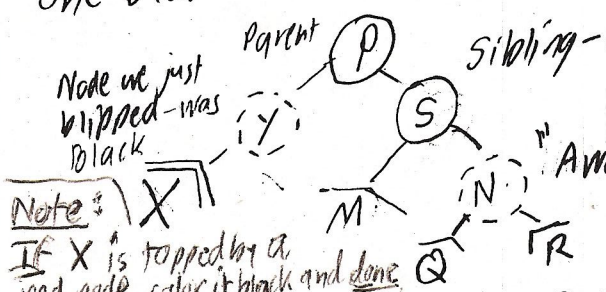
CSE 250 Lecture Wed 12/9/09 Fall 2009

Deletion in Red-Black Trees

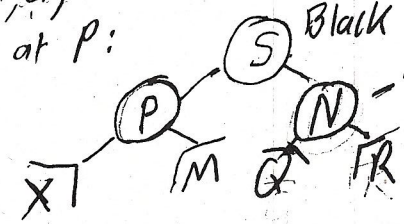
As with ordinary BST deletion, if the node is interior, then its inorder predecessor and successor are both leaves or "elbows" (one child). Choosing either, we move up its data and delete that node instead.

If the node to delete now is Red, we blip it as before. But if Black, after blipping it, we have a Subtree with One Black Too Few — a problem even if it's an empty subtree:



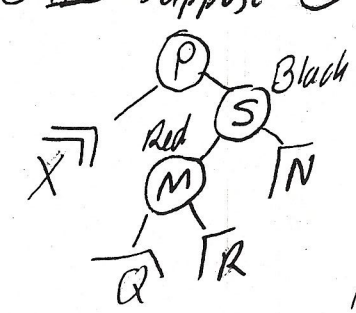
The double-line ⥅ means "This subtree Needs a Black". The big question is, does (S)ibling's subtree have a Black to give? Answer is Yes when it has a Red "Away-Leaning Niece": Then (S) itself is Black, so the subtree X and the subtrees rooted at M, Q, and R all have the same # b of Blacks. Now Rotate left at P:

(1)

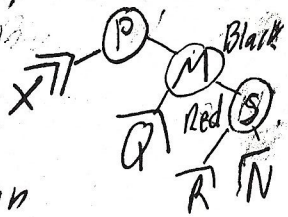


Text calls this Case 2 on p 688 (mirror image) when P is Red, but it works also when P is black — make N black too. Since X, M, Q, R all have same # of Blacks, and (S) gave what (Y) lost, we're done.

(2) Now suppose (S) has an In-Leaning Red Nephew:

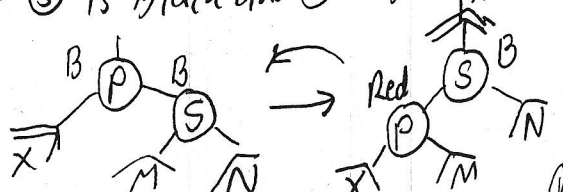


Again S is Black and X, Q, R, N have equal Black heights. Rotate right at S:



Then S itself becomes an Away Leaning Red Niece → prev case!

(3) Now suppose (S) has no red children. If (S) is Black and (P) is red, we swap their colors and are done. If (S) is Black and (P) is Black, we rotate left at P:

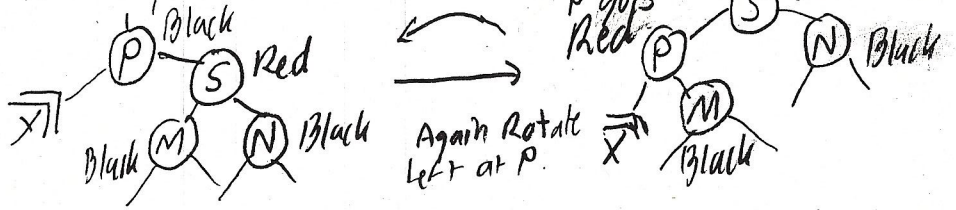


OK since X is not topped by a red node, norm.

If (S) now is the root of the whole tree, this is when its Black height goes down.

Now we recurse with (S) as new root asking for 1 more Black. Since (P) went Red that equalizes.

(4) So, we are left with the case where (S) is Red. Since (Y) was originally Black, (S) must have two Black children.



It seems like nothing has changed — the paths thru (M) and (N) still pick up 2 Blacks from the diagram while paths thru X are still missing 1 Black — but X took a step backwards. However, X now has a Black sibling (N) and another Black sibling awaits at node (N). Thus cases (1)-(3) will at worst push the recursion up to (S), so progress gets made! END