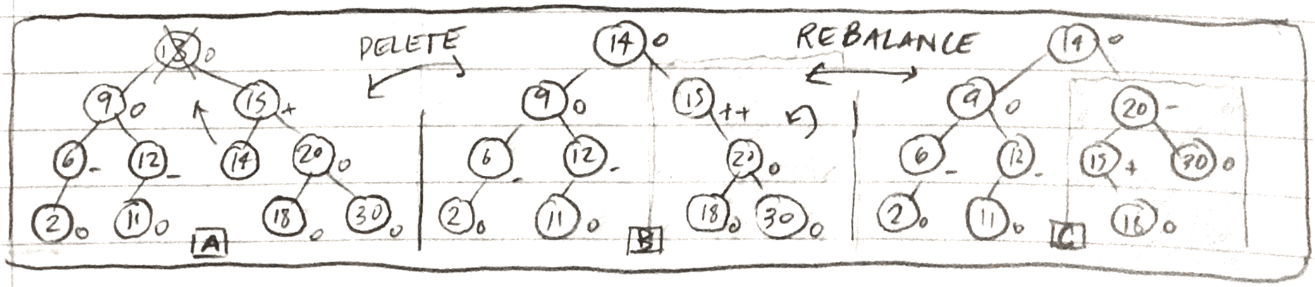
CSC263 Friday Tutorial Jan 26, 2018

- Delete from BST; DELETE(X)
  - (i) x has no children; trivial deletion — O(1)
  - (ii) x has one child; change one pointer — O(1)
  - (ii) x has 2 children; find successor, remove successor (recursive delete call), put successor value in x location — O(n) } height of BST not guarunteed log n

- Note: there will be questions that test your knowledge at the difference between BST and balanced trees

- Delete X from AVL tree:
  - find node k where x is stored, delete node x as in regular BST (outlined above), then rebalance the tree
  - in all three cases, we end up deleting a leaf;
    - (i) x has no children, x is leaf, delete leaf x
    - (ii) x has one child x'; x' must be leaf to preserve balance when x' replaces x, leaf x' is eliminated
    - (iii) x has two children, find successor (which by definition has at most one child), deleting successor is case (i) or case (ii)
  - every time deletion is made, go from deleted node to root and rebalance whenever unbalanced node encountered

Ex:



Ex: