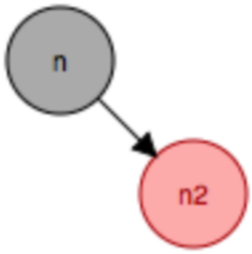
Consider a red-black tree formed by inserting n nodes with the algorithm described in class. Argue that is n > 1, the tree contains at least one red node.

Proof by induction:
Base case N = 2:



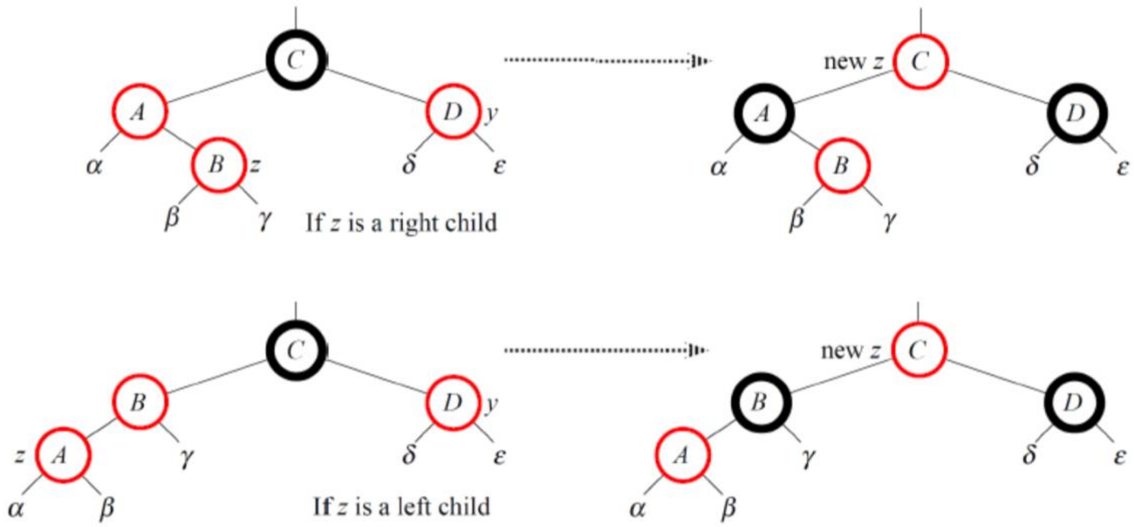
True.

Inductive Hypothesis:
Assume that for a red-black tree of n nodes, where 1<n<=N there must exist at least 1 red node.

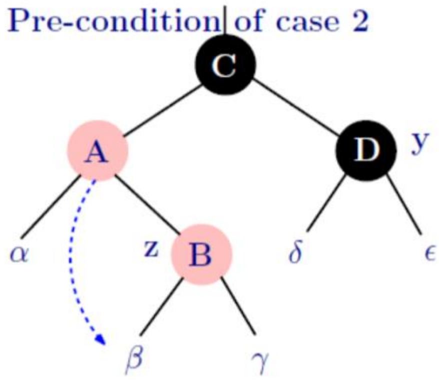
Inductive step:

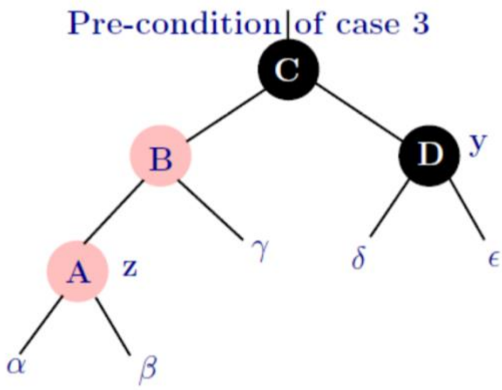
Case 1: node (n+1) that is inserted is the child of a black node. This is true from base case

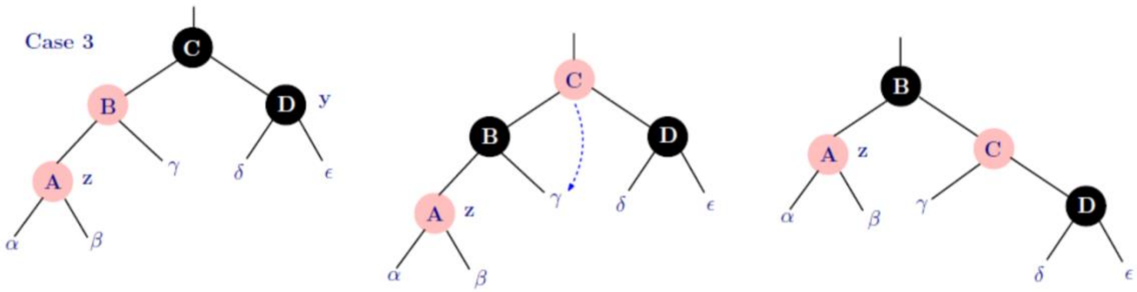
Case 2: node (n+1) that is inserted is the child of a red node.
In which case we must look at our insertion cases.



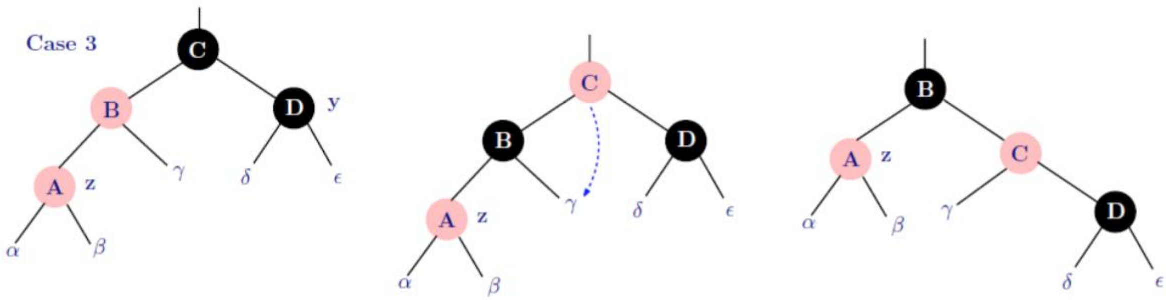
Insertion case 1: B remains red after the recoloring of the other nodes. It also remains the same color after we move reference B to Grandparent of B. Thus we have atleast 1 red node. Even after C is converted to black







Insertion case 2: The parent A of B remains red after the several rotation of B around A and then C. B remains red after recoloring A and C. Thus we have at least 1 red node.



Insertion case 3: A remains red after the rotation of B about C. B remains red after the recoloring of A and C. Thus we have atleast 1 red node.

In all cases we have at least 1 red node. Thus when n > 1, the tree contains at least one red node.