

c) A RBT will always have ~~red~~ red node for nodes greater than ~~as~~ initially. the root will be black.

Adding a second element to it regardless of its left or right position will ~~be~~ be a red node due to our implementation of algorithm (Base case).



Adding a third element there is two scenarios. One when its added on the same side which will result in both the leaves to be red or the other adding a red node to the ~~in~~ null side which will also result both leaves into red.

After this regardless of which side you add it on only the left or right subtree colors change which then proves that there will always be 1 red subtree.

Problem 9.2

1). Implemented on "RBT.cpp".