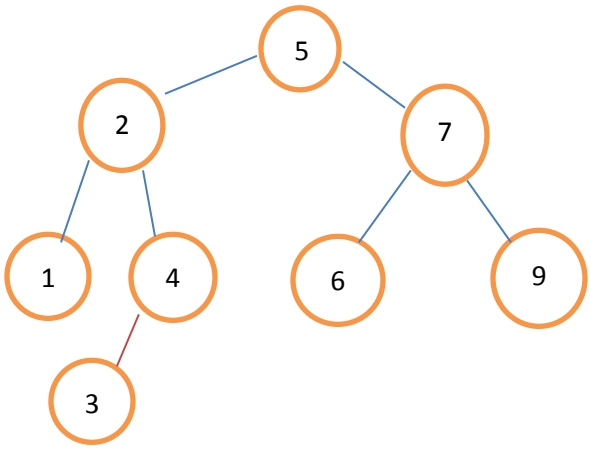
Mamoutou Sangare

CSC 226 Assignment 2

1 – Constructing a red-black tree by inserting the following keys in the order given: 2,1,4,5,9,3,6,7. The blue lines represent the black links. With the following keys, after reconstruction, all the links will be black except the one between 3 and 4




2- Prove that the height of a red-black tree with N nodes is at most 2 logN

Let x be a node in RBT and bh(x) be the black height of x. Then the length of a longest path from x to a leaf = $2bh(x)$ (a path on which red and black links alternate). Now the length of the the shortest possible path from x to a leaf is a path that only contains blacks $bh(x)$. Now consider any node x and the subtree rooted at x. We will first show that this subtree has at least $2^{bh(x)}$ internal nodes. We show that through induction on the height (h) of x.

Base Case:
If h(x) = 0, bh(x) = 0, there is only one internal nodes.
Hence, $2^0 = 1$

Induction Hypothesis:

Let h(x) > 0 and x1 and x2 be its two children
We know that h(x1), h(x2) are both $\leq$ h(x) . Let assume the result to be true for x1 and x2. Our goal is to show the result is true for x.
Induction Step:
bh(x1) $\leq$ bh(x) and bh(x) $\geq$ bh(x) -1
bh(x2) $\leq$ bh(x) and bh(x) $\geq$ bh(x) -1
Therefore, the tree with root x1 has at least $2^{bh(x) - 1}$ internal nodes and the tree with root x2 has

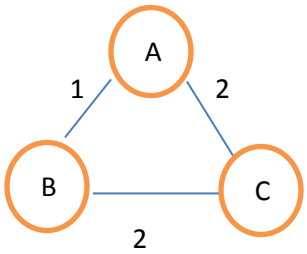
at least $2^{bh(x)-1}$ internal nodes. Thus the tree with root x has at least
$1 + 2^{bh(x)-1} + 2^{bh(x)-1} = 2^{bh(x)} + 1 \geq 2^{bh(x)}$ internal nodes .
Finally, let h be the height of the tree. Then
$bh(root) \geq \frac{h}{2}$
$N \geq 2^{(h/2)}$
$\log_2 N \geq \log_2 (2^{(h/2)})$
$\log_2 N \geq \frac{h}{2}$
$2 \log_2 N \geq h$ so $\log N \geq h$ as desire

3- Prove that the Reverse-Delete algorithm always outputs the MST (distinct weight)
Consider any edge e = (v, w) removed by the algorithm (e connects v and w). Just before the edge e is removed, it must be in on some cycle in the graph. (Otherwise removing it would disconnect the graph). As the first encountered on that cycle, it must be the most expensive (greatest weight) on it, so by the cycle property it doesn't belong to any minimum spanning tree. Hence the "Reverse Delete" algorithm produces a minimum spanning tree, because it removes only edges that cannot be in any minimum spanning tree.

4. Prove that if all the edges of a graph G have distinct weights, the MST of G is unique. We will use a proof by contradiction. Suppose there are two minimum spanning trees A and B. Let e be the edge and e is in only A. In other words, e is in A but not B. Now suppose B has the smallest weight. Also, suppose e is the edge uv. Then B must contain a path from u to v which is not the edge e. So if we add e to B, then we get a cycle. If all the other edges in the cycle were in A, then A would contain a cycle, which it cannot. So the cycle must contain an edge f not in A. Hence, by the definition of e (and the fact that we have distinct weights) the weight of f must be greater than the weight of e. So if we replace f by e we get a spanning tree with smaller total weight which is a contradiction. Therefore, if all the edges of a graph G have distinct weights, then the MST of G is unique.

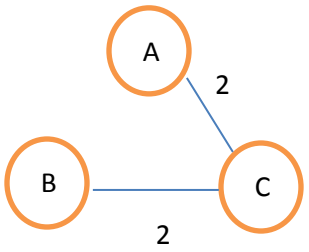
5- Give a counter example to show that MBST of G is not always a MST of G

Let G be:

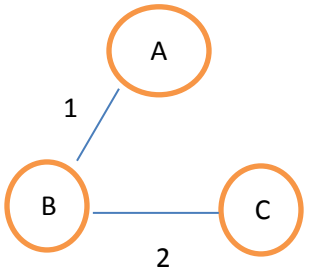


Now, let makes two MBST from the G (called T1 and T2).

T1:




T2:




Both T1 and T2 are MBST, but we can see that T1 is not a MST which is a counter example. Thus MBST of a graph G is not always a MST of G