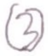
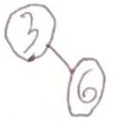
step 1: insert 3

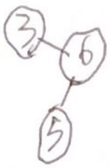


AVL

Step 2: insert 6

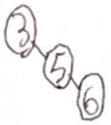


Step 3: insert 5

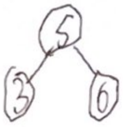


Step 3a: right-left rotation on 3

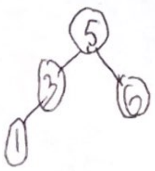
Step 3aa: right rotation on 6



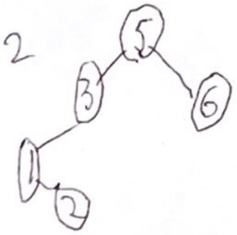
step 3ab: left rotation on 3



step 4: insert 1

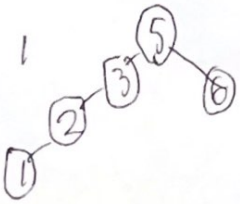


Step 5: insert 2

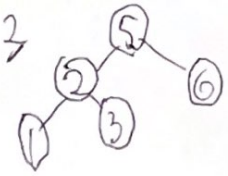


step 5a: left-right on 3

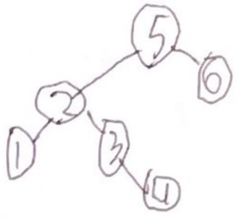
Step 5aa: left on 1



step 5ab: right on 3

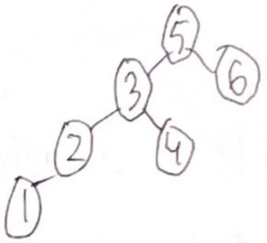


Step 6: insert 4

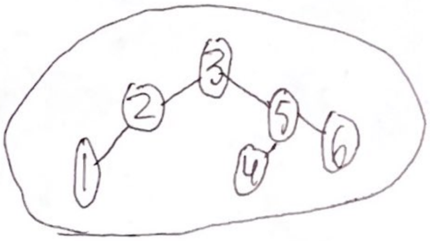


Step 6a: left-right on 5

Step 6aa: left on 2

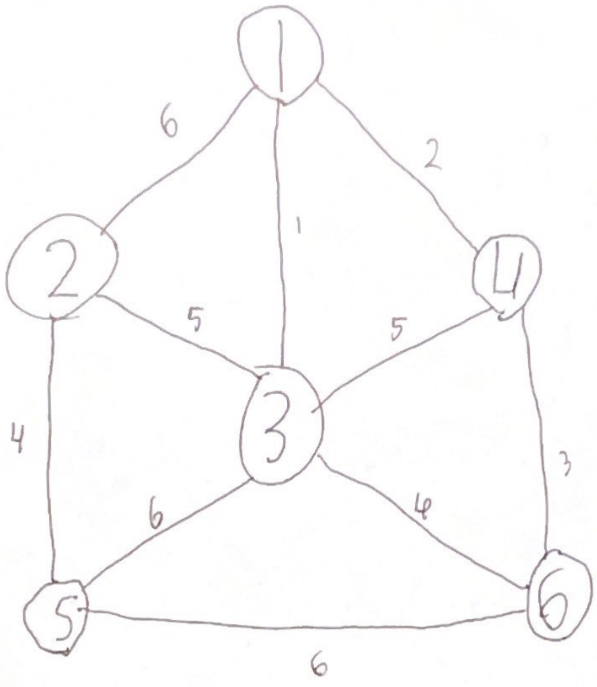


step 6ab: right on 5



↑

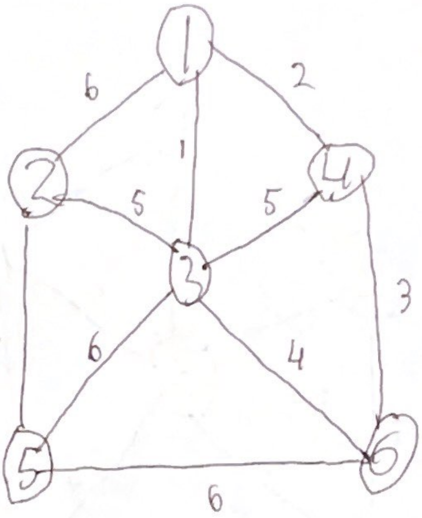
Making 4 5's left child since new root already had a right child



Starting w/ 1, we will go to 3 since the weight from 1 to 3 is 1 the smallest edge going to 1. From there we'll go from 1 to 4 since the edge between those 2 is the smallest available node going to 1 or 3 (excluding 1 to 3). Next I'll choose 4 to 6 since that's the next smallest edge going to the available nodes (except of 3). Then I'll go to 2 from 3, then 2 to 5.

(1,3) (1,4) (4,6) (3,2) (2,5)

Kruskai's algorithm



First, we'll put (1,3) in our list since it is the smallest edge. The next smallest edge is (1,4), then (4,6), so we'll add those. Initially, you add (3,6), but this actually forms a cycle between 4, 3 and 6, so we will ignore this edge. Next we'll add (5,2) since this has the next smallest weight, then finally (3,2).

(1,3) (1,4) (4,6) (5,2) (2,3)