CSE 2331 Homework 8

1. Imagine contracting the edge above each red node in a red-black tree, so that it shrinks to nothing, merging it with its parent. All other edges are maintained. What is the most that this operation can change the height of the tree? What is the maximum number of children of any node?

   Solution. We saw in class that since no two reds can be consecutive, just hunder half of the nodes on any simple root-leaf path can be red. Since contracting red nodes removes them, this reduces the height of the tree by at most half. There can be at most four children, corresponding to when a black node absorbs both of its red children who, in turn, both had two children. The result is a black node with edges to its former left and right grand-children. □

2. Bound, as tightly as possible, the number of internal (non-nil-leaf) nodes in red-black trees of black-height, h_b.

   Solution. Recall that the height of a red-black tree is its height as a tree. For consistency we will use the definition given in the book which is the number of nodes on the longest root to nil-leaf path. On the other hand, the black-height of a red-black tree is the number of black nodes on any root to leaf path, including the nil leaf nodes but excluding the root.

   Note: The *black-height* of empty trees is not defined, while their *height* is 0.

   Observe that by the property that every simple path from a node to a descendant nil leaf contain the same number of black nodes. Thus, the smallest all-black tree with black-height h_b is a full binary tree of all black nodes. Excluding the nil leaves there are at least $2^{(h_b-1)+1} - 1 = 2^{h_b} - 1$ internal nodes.

   On the other hand, the largest tree possible with black-height h_b is a complete (including last row) binary tree which interleaves h_b+1 black rows with h_b red ones in an alternating fashion, with the last row being all black nil leaves. The height of this tree, excluding the layer of nil leaves, is $h = 2h_b - 1$ and therefore contains $2^{(2h_b-1)+1} - 1 = 4^{h_b} - 1$ nodes. □

3. Insert nodes 1 to 8 into a red-black tree, draw the tree before and after any fixup operations. How does the result compare to what happens with INSERT for regular BSTs?

   Solution. See images at the end of this document. □

4. Let G be a weighted graph with unique edge weights containing a cycle, and let T be a minimum spanning tree of G. Argue that T is also minimum spanning tree of $G' = (V(G), E(G) - \{e\})$, where e is a cycle edge with maximal weight among all edges on the cycle.

   Solution. We argue that maximal weight cycle edges are not used in T and therefore removing them still results in a spanning tree of minimial cost.

   Consider any cycle in G and let e denote its maximal weight edge. Suppose, for contradiction, that T uses e_M. Note that e can be swapped with any other cycle-edge and the tree will remain conntected. Thus this edge can be substituted for one of strictly lower cost, resulting in a cheaper tree. It follows that T does not use this edge.

   It immedately follows that T also spans version of the graph that is the same as G except for excluding the edge, e. □

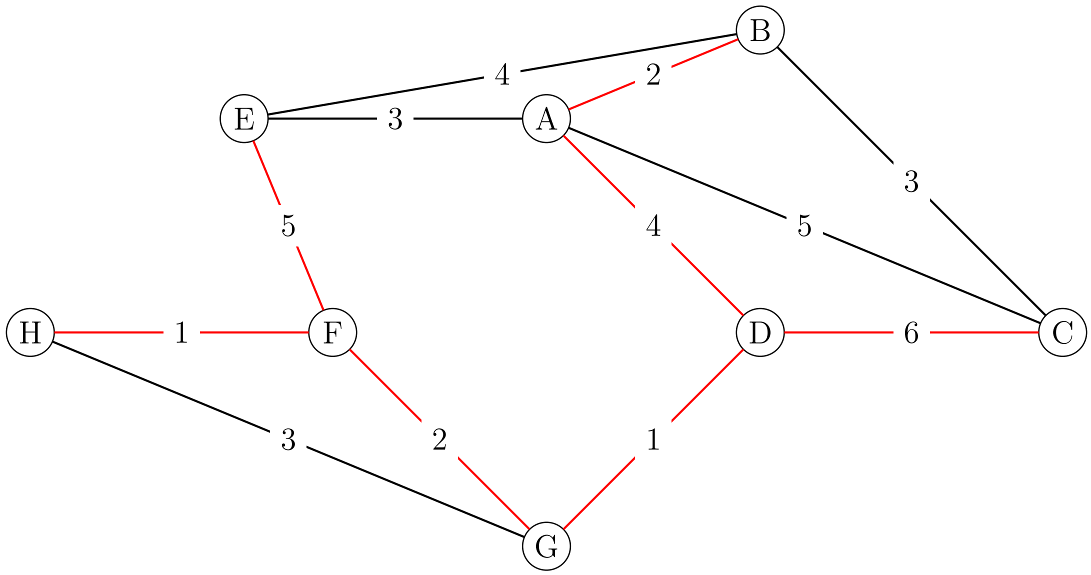
5. Argue that the minimum spanning tree is unique whenever the edge weights are unique. That is, whenever no two edges have the same weight. Hint: consider any cut of the graph.

   Solution. Consider any cut of the graph U, $V(G) - U$. For the moment assume there are many possible MSTs and fix one, T. Some edge of T must cross this cut or the graph is disconnected. Note that since T spans $V(G)$ it spans both U and $V(G) - U$. In other words, every node on both sides of the cut is in the tree. It follows that any edge crossing the cut results in a spanning tree. If any edge other than the minimum is selected the tree will not have minimal cost, since its cost could be reduced by using this edge. It follows that there is a unique choice of edge across any cut. Therefore, T is unique. □

6. Consider a graph with positive weight edges. Can the shortest path between any pair of vertices contain a cycle? Note we are talking about a cycle in a single path, not if the union of multiple shortest paths may form a cycle. (They can but this is irrelevant.)

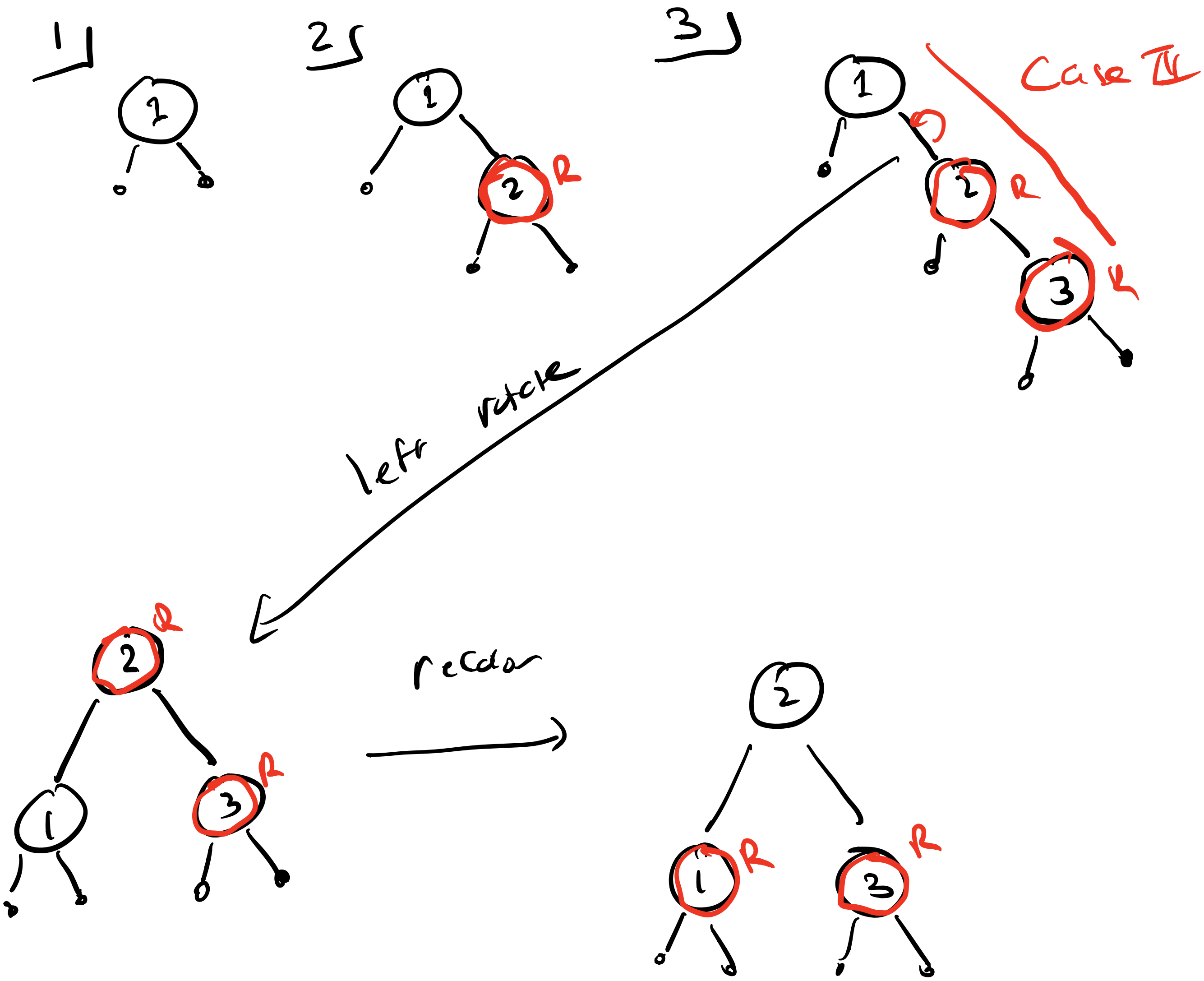
   Solution. It cannot. Suppose the shortest path from s to t contains a cycle. Then there is some vertex v which appears at least twice on the path from s to t. Deleting the a portion of the walk which loops back to v strictly decreases the length of the walk since the graph has positive edge weights. It follows that the shortest path between any pair of vertices cannot contain a cycle. □

7. Draw the shortest path's tree from H and label the nodes with their distances.

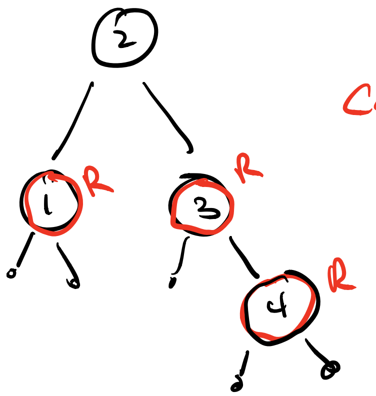


Solution. Due to some degeneracy there are a couple of different shortest-paths trees. For example, you could have the edge $H - G$ instead of $F - G$. However, regardless of which tree, the distances must be the same. You should have: $\delta(H, A) = 8$, $\delta(H, B) = 10$, $\delta(H, C) = 10$, $\delta(H, D) = 4$, $\delta(H, E) = 6$, $\delta(H, F) = 1$, $\delta(H, G) = 3$, $\delta(H, H) = 0$. $\square$

[redacted]

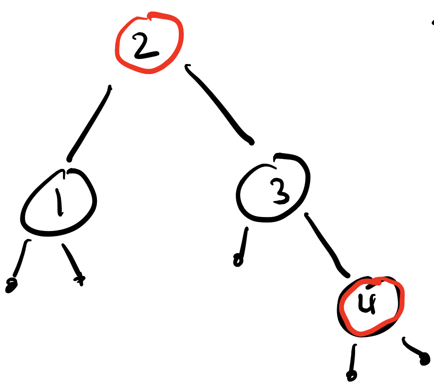


4)

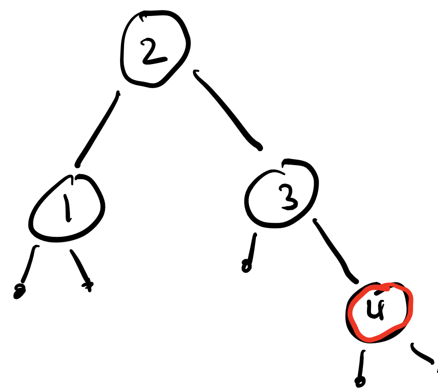


Case I

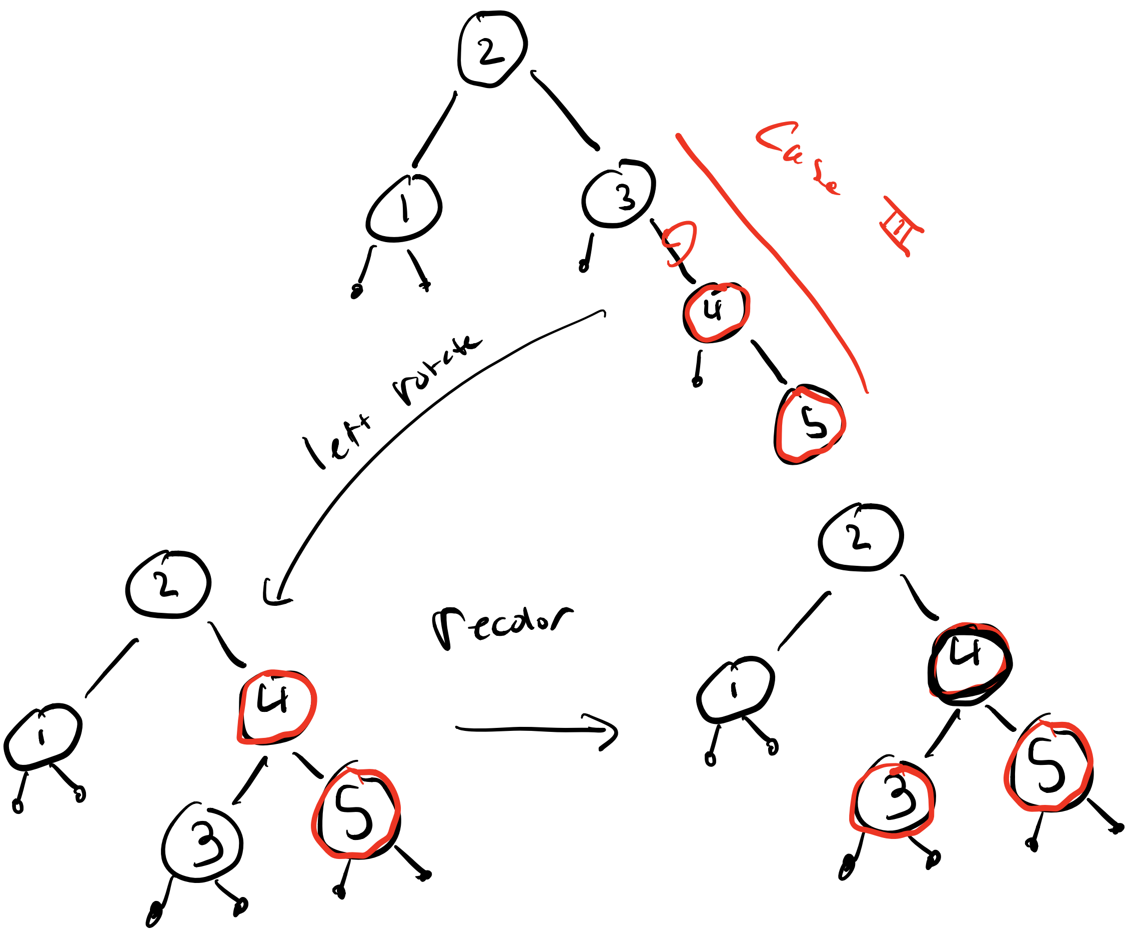
recolor



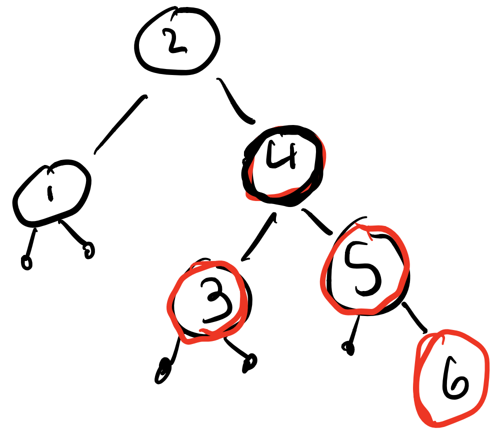
Set root Black



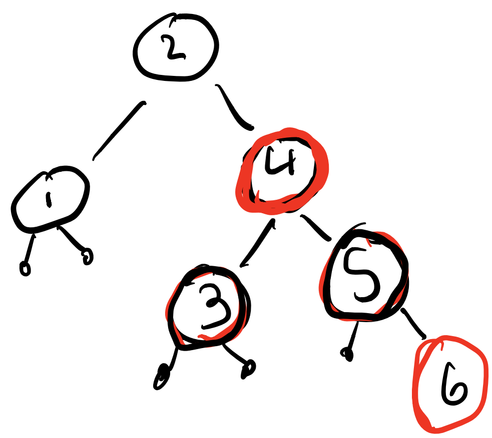
5)



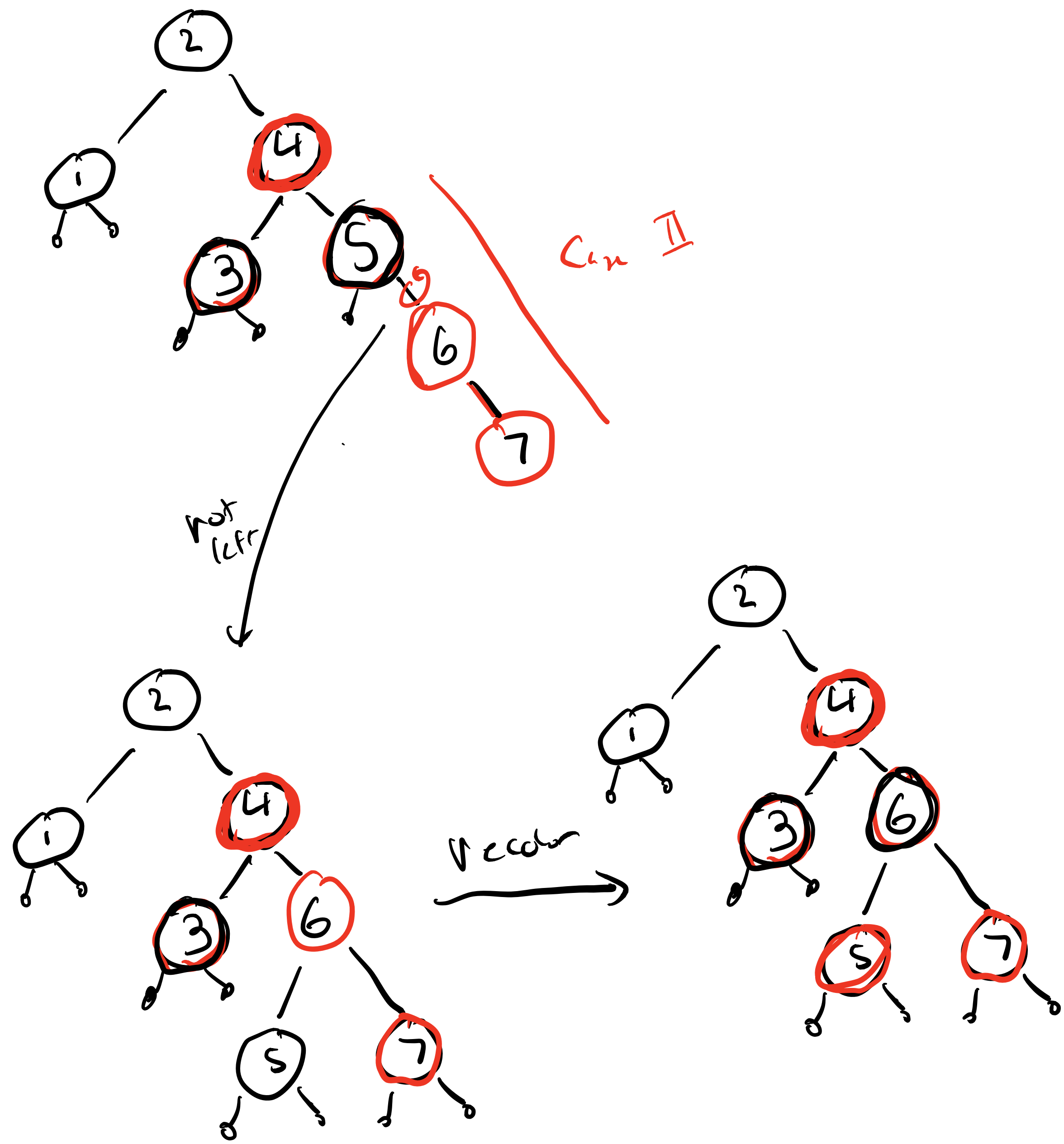
6)



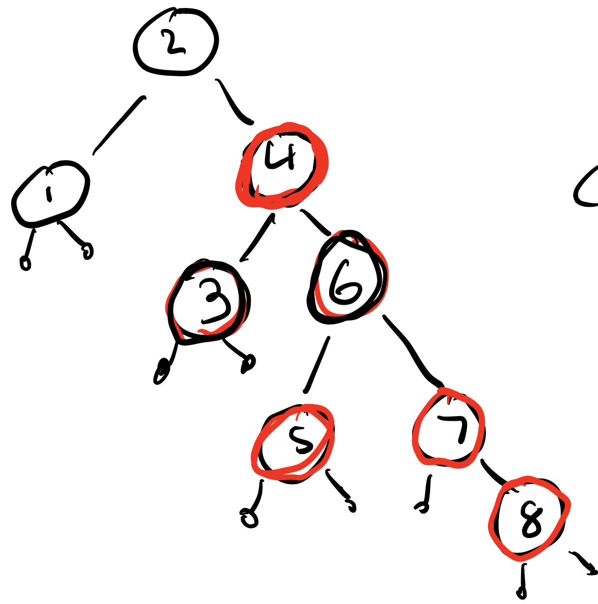
recolor



7]

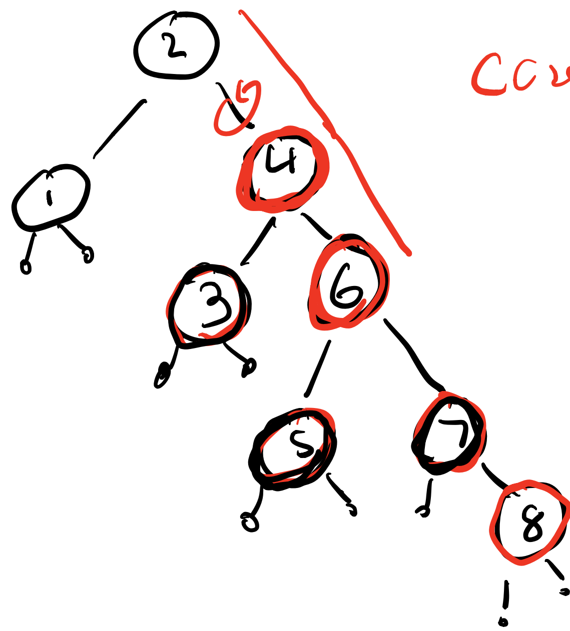


8)

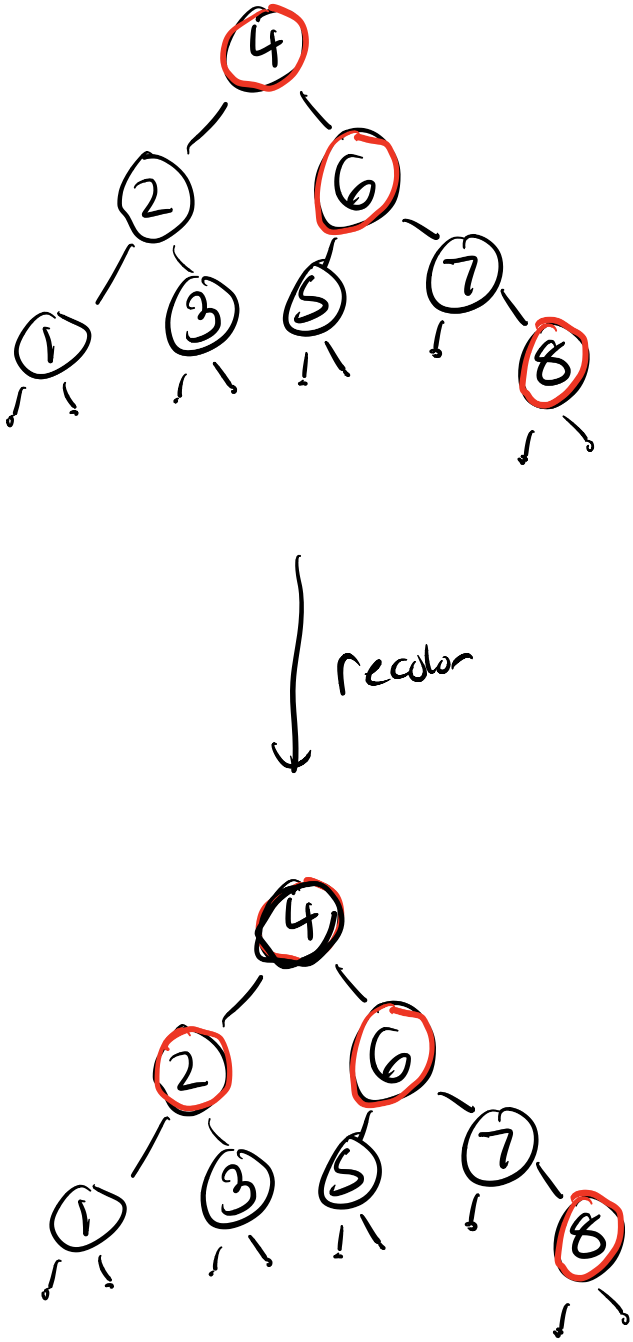


Case I

Recolor



Case III


4
2
6
1
3
5
7
8
recolor
4
2
6
1
3
5
7
8

BST looks like