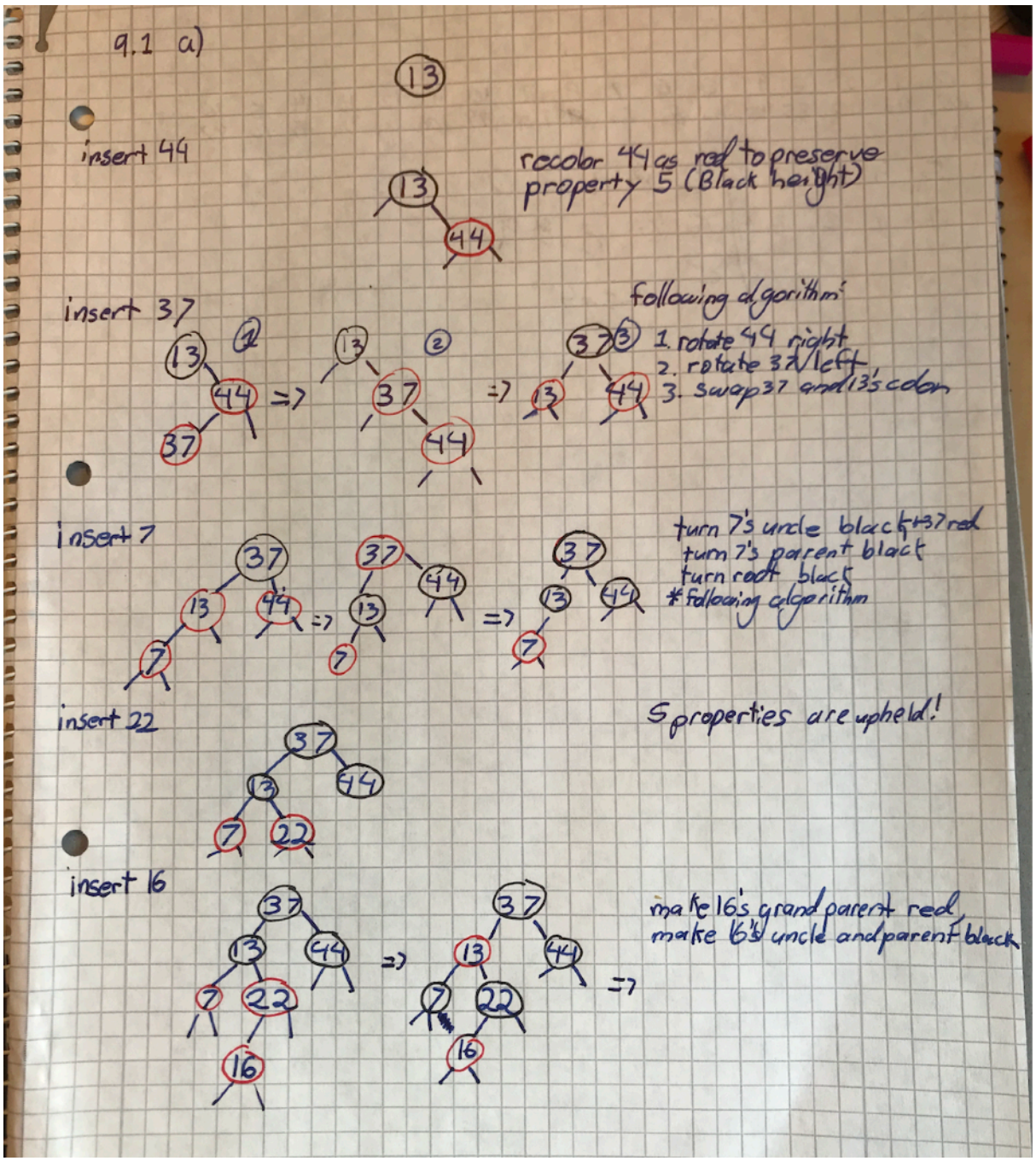
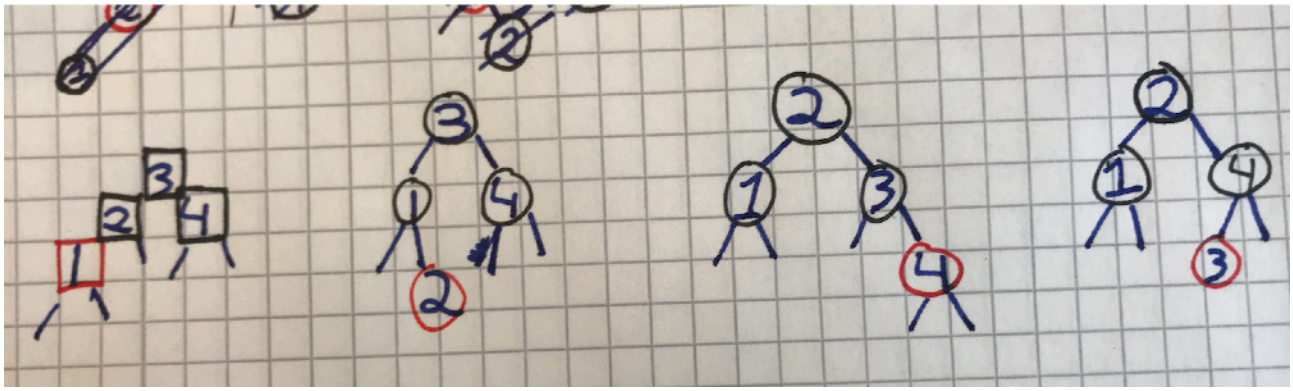

9.1 a)
13
insert 44
recolor 44 as red to preserve property 5 (Black height)
following algorithm:
1. rotate 44 right
2. rotate 37 left
3. swap 37 and 13's color
insert 37
insert 7
turn 7's uncle black, 37 red
turn 7's parent black
turn root black
* following algorithm
insert 22
5 properties are upheld!
insert 16
make 16's grandparent red
make 16's uncle and parent black

B



(c) Bonus (3 points) Consider a red-black tree formed by inserting n nodes with the algorithm described in the lecture slides. Prove that if n > 1, the tree contains at least one red node.

This is somewhat easy to prove by induction:

Base case: n = 2; this results in one red node and one black node.
Inductive step: assume there is at least one red node in any red black try of size n such that 1 < n < N.
Now it has to be shown that a tree with n+1 nodes has at least one red node. This can be done by going over the cases that the insertion algorithm goes over:
Case 1: the new element is inserted as a red node as the child of a black node, therefore, there is at least one red node in the tree (new element inserted is red).
Case 2: The uncle of the inserted node is also red: in this case, a color flip will be done, making the parent node and uncle nodes black and the grandparent node red. Even if this process is repeated iteratively going up the tree, at least one node in the tree will be red. Due to the color flip.
Case 3: The color of the new inserted node remains red after its parent and grandparent nodes are rotated and have their colours swapped.

Problem 9.2 Implementing Red Black Trees (16 points) Implement a red black tree (with integer nodes), closely following the specifications and algorithms from the lecture. Make sure you handle errors appropriately by printing messages or throwing exceptions. Your implementation has to be along the interface below with the following or equivalent components:
For full implementation, see RedBlackTree.h. and RedBlackTree.cpp. These implementations use templates, so for an integer red black tree, the integer template must be specified. The main.cpp program also shows some of the functionalities of the tree.

Sources:
https://www.geeksforgeeks.org/c-program-red-black-tree-insertion/
https://www.codesdope.com/course/data-structures-red-black-trees-deletion/
Course Slides
Tool for checking black height and violations referenced from: https://eternallyconfuzzled.com/red-black-trees-c-the-most-common-balanced-binary-search-tree