Advanced Data Structures and Algorithms
Quiz 1

For each of the below statements answer whether it is True or False. For each statement, you will get **2 mark** for the right answer and **-1** for the wrong answer. Write the answers on this sheet and return it back. You do not need to submit the rough sheets. Make sure to read through all the questions, as they are **NOT** arranged in ascending order of difficulty.

Name: **Roll No:**
Duration: 30 Minutes **Marks:**

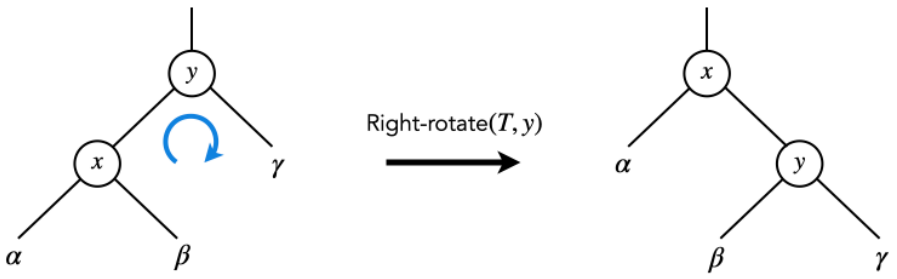
1. In any RB-tree, two siblings are always of the same colour.
False.

2. In the disjoint-set union data structure via trees implementation with only rank heuristic **not** path-compression, for every x, y, if **Union**(x, y) is called when x and y are in the same set S, then rank of the representative of S will not change.
False. Rank will increase by one.

3. Recall that in augmented RB-trees every node stores the number of nodes in the subtree rooted at that node. And we need to maintain this information during left and right rotations. In a right rotation the following is the correct way to maintain this information **after** the usual rotation operation in a tree:

i) $x.size = y.size$

ii) $y.size = y.right.size + y.left.size + 1.$



True. Note that these changes are done in the given order **after** doing rotations, that is, changing the pointers.

4. If a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child.

True.

5. If a directed graph with no cycles and real weights on edges, subpaths of a shortest path are also shortest paths.

True. The argument we saw in class will go through as there are no cycles.

6. In the disjoint-set union data structure via trees implementation with rank and path-compression heuristic that starts with 10 elements it is possible to have a tree of height 3 via a sequence of **Union**(x, y) operations.

True. Use **Union**(1,2), **Union**(3,4), **Union**(5,6), **Union**(7,8), **Union**(9,10), **Union**(2,4), **Union**(6,8), **Union**(4,8). Now, tree with representative 8 will have height 3.

7. The operation of insertion is commutative in the sense that inserting x and then y to a binary search tree leaves the same tree as inserting y and then x.

False. Inserting 1 then 2 initially will give a different tree than inserting 2 and then 1.

8. In the disjoint-set union data structure via trees implementation with only rank heuristic not path-compression, it is not possible to have a tree of height 1 containing all the elements via a sequence of **Union**(x, y) operations.

False. It is possible. Use **Union**(1,2), **Union**(2,3), **Union**(2,4), **Union**(2,5), …, **Union**(2,n), after **Make-Set** for every element.

9. Let v_1, v_2, u be three distinct vertices in a graph G. Then, if there exists a path from v_1 to u and a path from u to v_2, then there exists a path from v_1 to v_2 that contains u.

False. Consider an undirected graph $v_1 - v_2 - u$. We can also construct a counterexample for directed graphs. (Hint: Use cycles.)

10. In an RB-tree, for any node x, no path from x to the leaf is more than twice in length than any other path from x to the leaf.

True. We proved this in class.