**Lab 2**

**Name: Toka Alaa Elgindy ID: 14**

**Name: Nada Salama Mohammed ID: 55**

**Introduction:**

**1.1 Red Black Tree**

A red-black tree is a kind of self-balancing binary search tree in computer science. Each node of the binary

tree has an extra bit, and that bit is often interpreted as the color (red or black) of the node. These colors

bits are used to ensure the tree remains approximately balanced during insertions and deletions. Balance is

preserved by painting each node of the tree with one of two colors in a way that satisfies certain properties,

which collectively constrain how unbalanced the tree can become in the worst case. When the tree is

modified, the new tree is subsequently rearranged and repainted to restore the coloring properties. The

properties are designed in such a way that this rearranging and recoloring can be performed efficiently.

**1.2 Tree Map**

A Red-Black tree based Navigable Map implementation. The map is sorted according to the natural ordering

of its keys, or by a Comparator provided at map creation time, depending on which constructor is used. This

implementation provides guaranteed log(n) time cost for the containsKey, get, put and remove operations.

Algorithms are adaptations of those in Cormen, Leiserson, and Rivest’s Introduction to Algorithms.

**Data structure:**

1) Every element in the red-black tree is represented by a node which implements "INode" interface.

2) Tree Map is based on the red-black tree.

**Algorithm used:**











