PROJECT-4 REPORT

Analysis of Algorithms

Emre KÖSE – 150130037

23.12.2017

BLG 335E - Analysis of Algorithms I, Fall 2017
Project 4 – Report

1. **Briefly explain what you would do to correctly update the name of a person as a node in the Red– Black Tree.**

Our red-black tree is created using the insertion sort according to the key value that is the name of person. If we want to update the name of a person, which has already in the our red-black nodes, we cannot do it with brute-force strategy. Because this event can cause a faulty tree like the updated node can have a lower key value than its' parents although it had a higher key value before the updating operation. This issue cannot fixed with any maintenance in insertion. Therefore, firstly, the node should be found in the tree and the attributes those will not be changed of node should be kept in temporary object. After that, the node, which will be updated, should be deleted and the new node that has a new key value (name) should be inserted to our red-black tree with the attributes those are saved before.

To sum up, while we update the name of the person, actually, we are changing the key value of the node and this can cause whole tree to change. Hence, we find the node and delete it, then adding a new node with the intended name. At last, we should use the insertion maintenance function to handle rotation and coloring issues.

2. **Briefly explain what you would do to correctly increment (by 1) the ages of all people in the Red– Black Tree.**

The age attribute is not used as a key in insertion. In addition, this is not used in the methods these find the n^{th} man and n^{th} woman nodes. Therefore, the changing of age will not affect any node order in the tree. We can search the name in the tree with traversing and after finding the node we can we can increment the value of age. After that, there will no need to any update because the key values were not changed and the number of men and women are same as before.