**Data Structures**

**Spring 2020**

**Written Homework**

**Date: 2020. 05. 11**

**ID/Name: 201533661 이승수**

**[HW(RBtree)]**

**Red-Black Tree Insert and Delete. (20 points)**

**-Insert: Cat Dog Bat Fish Chicken Cow Tiger Eagle Lion Snake Bird Owl Mouse**

**-Delete: Cow Snake Owl Cat Mouse Eagle Bird**

**Notation:**

**N(new node), P(Parent), G(Grand parent), U(uncle)**

**[Insertion Case]**

**(1) N**is the root node

**(2) N'**s parent (**P**) is black

**(3) N'**s parent (**P**) and uncle (**U**) are red

**(4) N**is added as the right child of the left child of grandparent, or**N**is added as the left child of the right child of grandparent (**P**is red and**U** is black)

**(5) N**is added as the left child of the left child of grandparent, or**N**is added as the right child of the right child of grandparent (**P**is red and**U** is black)

**[Insertion Phase]**

|  |  |  |
| --- | --- | --- |
| **Insert Cat** | **Insert Dog** | **Insert Bat** |
|  |  |  |

|  |
| --- |
| **Insert Fish** |
|  |

|  |
| --- |
| **Insert Chicken** |
|  |

|  |
| --- |
| **Insert Cow** |
|  |

|  |  |
| --- | --- |
| **Insert Tiger** | **Insert Eagle** |
|  |  |

|  |
| --- |
| **Insert Lion** |
|  |

|  |
| --- |
| **Insert Snake** |
|  |

|  |
| --- |
| **Insert Bird** |
|  |

|  |
| --- |
| **Insert Owl** |
|  |

|  |
| --- |
| **Insert Mouse** |
|  |

**Notation:**

**N(new node), P(Parent), G(Grand parent), U(uncle), C(Child), D(Node to delete)**

**[Deletion Case]**

If the deleted node was red, we are done.

If a black node is deleted and replaced by a black child, the child is marked as a double black (or extra black) node.

**(Case 1) D is a red node**

Simply replace it with its child **C**

C must be black by property 4.

**(Case 2) D is black and C is red.**

Repaint **C** black

Simply removing a black node could violate properties 4 and 5.

**(Case 3) P, S, and S's Children Are Black**

Simply repaint **S** red.

The result is that all paths passing through**S**, which are precisely those pathsnotpassing through **N**, have one less black node.

**(Case 4) S and S's Children Are Black, But P Is Red.**

Simply exchange the colors of **S** and **P**.

**(Case 5) S is black,S's right child is black**

Rotate right at **S**, so that **S'**s left child becomes **S'**s parent and **N'**s new sibling. We then exchange the colors of**S** and its new parent.

**(Case 6) S is black, S's right child is red**

Rotate left at **P**, so that **S** becomes the parent of **P** and **S'**s right child. We then exchange the colors of**P**and**S**, and make**S'**s right child black.

If a path does not go through**N**, then there are two possibilities:

It goes through**N'**s new sibling. Then, it must go through**S**and**P**, both formerly and currently, as they have only exchanged colors and places. Thus the path contains the same number of black nodes.

It goes through**N'**s new uncle,**S'**s right child. Then, it formerly went through**S**,**S'**s parent, and**S'**s right child (which was red), but now only goes through**S**, which has assumed the color of its former parent, and**S'**s right child, which has changed from red to black (assuming**S'**s color: black). The net effect is that this path goes through the same number of black nodes.

**[Deletion Phase]**

|  |
| --- |
| **Delete Cow** |
|  |

|  |
| --- |
| **Delete Snake** |
|  |

|  |
| --- |
| **Delete Owl** |
|  |

|  |
| --- |
| **Delete Cat** |
|  |

|  |
| --- |
| **Delete Mouse** |
|  |

|  |
| --- |
| **Delete Eagle** |
|  |

|  |
| --- |
| **Delete Bird** |
|  |