**Data Structure Written HW#3**

**Spring 2020**

**Student ID/Name: 201533661 이승수**

**Due Date: 2020. 04. 22.**

**HW#3-1. AVL Programs (30 points)**

**Hand trace the Above program in doing the following sequence of 5 inserts:**

**(for the purpose of easier conceptualization, assume the node key is a string, and node address is an integer, as shown below)**

* **fish (address: 100), dog (addr: 120), cat (addr: 130), mouse (addr: 140), lion (addr: 150)**

**[Answer]**

**1. <insert fish>(initialization)**

**Insert**(\*parent [=NULL], element x [= fish(100)], \*unbalanced[= ?]) /\*1st avl start\*/

{

!\*parent [=TRUE]

/\*avlInsert:type 1\*/

\*unbalanced [= TRUE]

[\*parent is allocated by malloc()]

(\*parent)->leftChild [= NULL], (\*parent)->rightChild [= NULL]

(\*parent)->bf [= 0]

(\*parent)->data [= fish]

} **function end**/\*1st avl end\*/

**2. <insert dog>**

**Insert**(\*parent[ = fish(100)], element x [= dog(120)], \*unbalanced[ = TRUE]) /\*2nd avl start\*/

{

x.key [= 120], (\*parent)->data.key[ = 100]

(x.key > (\*parent)->data.key)[=TRUE]

/\*avlInsert:type 3\*/

**avlInsert**((\*parent)->rightChild [= NULL], element x [= dog(120)], \*unbalanced [= TRUE)])

{ /\*2nd-1 avl start\*/

!\*parent [= TRUE]

/\*avlInsert:type 1\*/

\*unbalanced [= TRUE]

[\*parent is allocated by malloc()]

((\*parent)->rightChild )->leftChild [= NULL],((\*parent)->rightChild )->rightChild [= NULL]

((\*parent)->rightChild )->bf [= 0]

((\*parent)->rightChild )->data [= dog]

} **function end**/\*2nd-1 avl end\*/

\*unbalanced [= TRUE]

((\*parent) [= fish(100)])->bf[ = 0]

**case 0:**

fish(100)->bf [= -1]

} **function end**/\*2nd avl end\*/

**3. <insert cat>**

**Insert**(\*parent [= fish(100)], element x [= cat(130)], \*unbalanced [= TRUE]) /\*3rd avl start\*/

{

x.key [= 130] , (\*parent)->data.key [= 100]

x.key > (\*parent)->data.key[=TRUE]

/\*avlInsert:type 3\*/

**avlInsert**(((\*parent)->rightChild)[ = dog(120)], element x [= cat(130)], \*unbalanced [= TRUE])

{ /\*3rd-1 avl start\*/

/\*avlInsert:type 3\*/

**avlInsert**(((dog(120)->rightChild)[=NULL],element x[=cat(130)],\*unbalanced[= TRUE])

{ /\*3rd-2 avl start\*/

(((\*parent)->rightChild) ->rightChild )[= TRUE]

/\*avlInsert:type 3\*/

{

\*unbalanced [= TRUE]

[\*parent is allocated by malloc]

(((\*parent)->rightChild) ->rightChild)->leftChild [= NULL]

(((\*parent)->rightChild) ->rightChild )->rightChild [= NULL]

(((\*parent)->rightChild) ->rightChild)->bf = 0

(((\*parent)->rightChild) ->rightChild)->data = cat

}

} **function end** /\*3rd-2 avl end\*/

\*unbalanced [= TRUE]

(\*parent)[= dog(120)]->bf [= 0]

**case 0:**

((\*parent)->rightChild)[=dog(120)]->bf[ = -1]

} **function end**/\*3rd-1 avl end\*/

Unbalanced[=TRUE]

((\*parent) [=fish(100)]->bf)[= -1]

**case -1:**

**rightRotation**(\*parent [= fish(100)], unbalanced[ = TRUE])

{ /\*rightRotation started\*/

child [= dog(120)]

(dog(120)->bf) [= -1]

{ /\*RR\*/

(\*parent[=fish]->rightChild)[=NULL] /\*Cut the link between \*parent and child\*/

(child->leftChild)[=NULL] = \*parent [=fish(120)]

(\*parent [=fish(120)])->bf [= 0]

\*parent [= dog(120)]

}

(\*parent[=dog])->bf [= 0]

\*unbalanced [= FALSE]

} **function end** /\*rightRotation end\*/

} **function end**/\*3rd avl end\*/

**4. <insert mouse>**

**Insert**(\*parent = dog(120), element x = mouse(140), \*unbalanced = TRUE)

{ /\*4th avl start\*/

x.key [= 140] , (\*parent)->data.key [= 120]

(x.key > (\*parent)->data.key)[=TRUE]

/\*avlInsert:type 3\*/

**avlInsert**(((\*parent)->rightChild)[= cat(130)], element x = mouse(140), \*unbalanced = TRUE)

{ /\*4th-1 avl start\*/

!\*parent = TRUE

/\*avlInsert:type 3\*/

**avlInsert**((cat(120)->rightChild)[ = NULL], element x = mouse(140), \*unbalanced = TRUE)

{ /\*avlInsert:type 3\*/

\*unbalanced[ = TRUE]

[\*parent is allocated by malloc()]

((cat(120)->rightChild )->leftchild)[ = NULL]

((cat(120)->rightChild) ->rightchild)[= NULL]

(\*parent->bf)[ = 0]

(\*parent->data)[= mouse]

} **function end**

\*unbalanced[ = TRUE]

((\*parent)[ = cat(130)]->bf) [= 0]

**case 0:**

cat(130)->bf = -1

} **function end** /\*4th-1 avl end\*/

((\*parent) [= dog(120)]->bf) [= 0]

**case 0:**

(dog(120)->bf) [= -1]

} **function end** /\*4th avl end\*/

**5. <insert lion>**

**Insert**(\*parent [= dog(120)), element x [= lion(150)], \*unbalanced [= TRUE])

{ /\*5th avl start\*/

x.key [= 150] , ((\*parent)->data.key)[ = 120]

( x.key > (\*parent)->data.key)[=TRUE]

{ /\*avlInsert:type 3\*/

**Insert**((\*parent)->rightChild = cat(130), element x = lion(150), \*unbalanced = TRUE)

{

**Insert**((\*parent)->rgihtChild[=mouse(140)],element x[= lion(150)],\*unbalanced[= TRUE])

{ /\*avlInsert:type 1\*/

!\*parent[ = TRUE]

**Insert**((mouse->rightChild)[= NULL],element x[= lion(150)], \*unbalanced [= TRUE]

{

\*unbalanced[ = TRUE]

[\*parent is allocated by malloc()]

\*parent->leftchild [= NULL, NULL->rightchild [=NULL]

\*parent->bf [= 0]

\*parent->data [= lion]

} **function end**

\*unbalanced[ = TRUE]

( (\*parent) = mouse(140)->b)f[= 0]

**case 0:**

mouse(140)->bf = -1

} **function end**

(\*parent) cat(130)->bf = -1

case -1:

**rightRotation**(\*parent = cat(130), unbalanced = TRUE)

{

(child = cat(130)->rightChild )[= mouse(140)]

(child->bf) [= -1]

{

(cat->rightChild (mouse)) [= NULL]

(child->leftChild)[= \*parent (cat)]

((\*parent (cat))->bf )[= 0]

\*parent[ = child (mouse)]

}

((\*parent(mouse))->bf) [= 0]

\*unbalanced[ = FALSE]

} **function end** /\*rightROtation end\*/

} **function end** /\*5th avl end\*/

**HW#3-2. AVL Tree Insert and Delete (10 points)**

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

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

**[Answer]**

**[Insert Phase]**

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

|  |  |
| --- | --- |
| Insert Fish | Insert Chicken |
|  |  |

|  |  |
| --- | --- |
| Insert Cow | |
| **RR**  **LR** | |
| Insert Tiger |
| **RR** |

|  |  |
| --- | --- |
| Insert Eagle | Insert Lion |
|  |  |

|  |
| --- |
| Insert Snake |
| **RL**    **LR** |

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

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

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

**[Delete Phase]**

|  |
| --- |
| Delete Cow |
| **Delete Cow**  **Merge**    **LL** |

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

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

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

|  |
| --- |
| Delete Mouse |
| **Delete Mouse** |
| Delete Eagle |
| **Delete Eagle**    **LL** |
| Delete Bird |
| **Delete Bird** |