Hw11

소프트웨어융합학과 2019102106 예다정

1. 만약 트리가 complete tree라면 중간에 비어 있는 노드, 즉 array에 사이사이에 dummy values가 들어갈 일이 없기 때문에 array-based queue로 구현하는 것이 편하고 트리 중간에 비어 있는 노드가 여러 개라면 linked queue로 구현하는 것이 더 좋다.

2.

a. B – D – J – K – M – N – P – Q – R – T – W – Y

b. B – J – D – N – P – M – K – R – W – Y – T – Q

c. Q – K – D – B – J – M – P – N – T – R – Y – W

3.

bool TreeType::Ancestor(TreeNode\* tree, ItemType value)

{

bool found = false;

if (tree == NULL)

return false;

if (tree->info == value)

return true;

if (value < tree->info)

found = Ancestor(tree->left, value);

else

found = Ancestor(tree->right, value);

if (found)

cout << tree->info << endl;

return found;

}

4.

a, b.

void Insert(TreeNode\*& tree, ItemType item);

TreeType TreeType::MirrorImage()

//호출 트리에서 각 노드의 left, right가 뒤바뀐 새로운 트리를 반환한다.

//pre: 트리가 초기화돼있어야 한다.

//post: 트리의 좌, 우 노드가 바뀐 새로운 트리가 생성되고 반환된다.

{

TreeType mirrorTree(\*this); //copy constructor

ResetTree(POST\_ORDER);

ItemType item;

bool finished = false;

while (!finished)

{

GetNextItem(item, POST\_ORDER, finished);

mirrorTree.MirrorInsertItem(item);

}

return mirrorTree;

}

void TreeType::MirrorInsertItem(ItemType item)

// Calls recursive function Insert to insert item into tree.

{

MirrorInsert(root, item);

}

void MirrorInsert(TreeNode\*& tree, ItemType item)

// Inserts item into tree.

// Post: item is in tree; search property is maintained.

{

if (tree == NULL)

{// Insertion place found.

tree = new TreeNode;

tree->right = NULL;

tree->left = NULL;

tree->info = item;

}

else if (item > tree->info) // 부모 노드보다 큰 item을 왼쪽에 insert

Insert(tree->left, item); // Insert in left subtree.

else

Insert(tree->right, item); // Insert in right subtree.

}

c. binary searching이 가능하다. 왜냐하면 midPoint값을 중심으로 더 큰 item은 왼쪽에, 더 작은 item은 오른쪽에 존재하기 때문이다.

5.

5의 right child 위치에 insert된다.

6.

5 혹은 6

7.

a. free: 8 root: 0

|  |  |  |  |
| --- | --- | --- | --- |
| [0] | Q | 1 | 2 |
| [1] | L | 3 | 4 |
| [2] | W | 5 | ? |
| [3] | F | ? | ? |
| [4] | M | 6 | ? |
| [5] | R | ? | 7 |
| [6] | N | ? | ? |
| [7] | S | ? | ? |
| [8] | ? | 9 | ? |
| [9] | ? | ? | ? |

b. free: 9 root: 0

|  |  |  |  |
| --- | --- | --- | --- |
| [0] | Q | 1 | 2 |
| [1] | L | 3 | 4 |
| [2] | W | 7 | ? |
| [3] | F | 8 | ? |
| [4] | M | 6 | ? |
| [5] | ? | ? | ? |
| [6] | N | ? | ? |
| [7] | S | ? | ? |
| [8] | B | ? | ? |
| [9] | ? | 5 | ? |

8.

a. b, d, e

b. b, e

9.

a. dummy value로 -1을 쓸 수 있다.

b.

|  |  |
| --- | --- |
| num | 10 |
| [0] | 26 |
| [1] | 14 |
| [2] | 38 |
| [3] | 1 |
| [4] | -1 |
| [5] | 33 |
| [6] | 50 |
| [7] | -1 |
| [8] | 7 |
| [9] | -1 |
| [10] | -1 |
| [11] | -1 |
| [12] | 35 |
| [13] | 44 |
| [14] | 60 |
| [15] |  |

10.

a. false 🡪 [64~84]

b. false 🡪 has ‘two’ children

c. false 🡪 The ‘left’ child

d. true

e. false 🡪 ‘six’ levels that are full, and one additional level that contains some elements