Report

데이터 구조 설계

Project1

컴퓨터정보공학부 2017202037 오민혁

1. 문제 설명

Binary Search Tree를 이용하여 축구 구단 관리 프로그램을 구현하는 문제이다. 각 선수들의 명단과 정보는 ShootForLog.txt 에 저장되어 있다. 정보에는 포지션과 이적료, 이름, 능력치가 있다. 포지션은 Forward, Midfielder, Defender, Goalkeeper 로 총 4개이다. 이적료와 능력치는 정수 값으로 저장되어 있다. 구단주에게 받은 예산 안에서 ShootForLog.txt에 있는 정보들을 통해 가장 강력한 팀을 꾸려야 한다. 가장 강력한 팀의 정의는 다음과 같다.

1. 팀은 Forward 1명, Midfielder 1명, Defender 1명, Goalkeeper 1명 총 4명의 선수로 구성된다.
2. 4명의 선수의 이적료 합이 구단주에게 받은 예산과 같거나 작아야 한다.
3. 4명의 선수의 능력치의 합(=강력함)이 주어진 예산으로 구성할 수 있는 팀의 능력치의 합 중 가장 높아야 한다. 만약 능력치가 같다면 이적료의 합이 더 적은 쪽이 강력한 팀이 된다. 능력치의 합과 이적료의 합 모두 같다면 두개의 팀 모두 정답이다.

2. 스테이지별 알고리즘

Stage1. Setup

프로그램 실행 시 ShootForLog.txt의 정보들을 읽어서 포지션(Forward, Midfielder, Defender, Goalkeeper)에 따라서 각 선수들의 데이터를 능력치를 기준으로 서브 트리를 생성하여 저장 한 4개의 Binary Search Tree를 구성한다. Binary Search Tree의 특성 상 삽입하고자 하는 노드의 능력치가 루트 노드의 능력치보다 크면 오른쪽 서브 트리에 삽입해야 하고, 적으면 왼쪽 서브 트리에 삽입해야 한다. 루트 노드는 ShootForLog.txt에서 포지션 별로 가장 처음 적혀 있는 선수가 된다. 이를 구현하기 위해서는 삽입을 실행해 줄 함수 그리고 ShootForLog.txt에 있는 자료들을 tokenizing하여 BST에 저장해줄 함수가 필요하다.

- **Tokenizing 하여 BST에 Data들을 저장해 주는 함수 pseudo code**

```
Void tokenizing
        for(get each line in file))
                vector v
```

istringstream ss

for ',' is exist

if line[0] is space remove erase

put in data at vector

Insert according to position at each vector

이 함수는 Vector를 사용하여 문자 ','별로 tokenizing하여 각 포지션별로 구성되어 있는 BST에 data를 저장한다.

- **Insert 함수 pseudo code**

```
Void insert(data)
    TreeNode * InsertNode = new TreeNode(data)
    TreeNode * CurNode = root Node
    If (CurNode = NULL) root node is InsertNode
    Else
        infinite loop
            if (InsertNode->ability > CurNode -> ability)
                if(RightNode is exist)
                    CurNode->RightNode
                Else
                    RightNode=InsertNode
                    Break
            Else(CurNode->ability>InsertNode->ability)
                If(LeftNode is exist)
                    CurNode->LeftNode
                Else
                    LeftNode=InsertNode
```

Break

이 함수는 삽입 할 자리를 찾아줄 노드를 하나 생성하여 뿌리 노드로 설정해주고, 삽입 할 노드를 만든다. 그 다음 무한 루프 안에서 삽입 할 노드의 능력치가 현재의 노드보다 작으면 왼쪽으로, 크면 오른쪽으로 가는 알고리즘으로 위치를 찾아서 노드를 삽입한다.

Stage2. Print Players

포지션 당 하나씩 총 4개의 Binary Search Tree에 Player들의 data를 저장한다. 그리고 이 4개의 Binary Search Tree를 중위순회 하며 선수들의 data들을 출력한다. 순서는 공격수, 미드필더, 수비수, 골키퍼순이다.

- **중위순회(inorder)하며 data들을 출력해 줄 함수**

```
Friend ostream& operator<<(os, tree)
    Stack s
    TreeNode * Current = root
    Infinite loop
        for Current is not 0
            push s at current node and move at left
        if stack s not 0
            current = top of stack and print current, move at right
        else break
```

이 함수는 스택을 만들어 현재 노드 위치를 tree의 root로 설정해준다. 그리고 데이터들을 스택에 순서대로 저장해주면서 tree의 가장 왼쪽 노드로 이동한 후 스택에 더 이상 데이터가 없을 때 까지 스택의 가장 위에 있는 데이터를 현재 노드로 설정하고 출력한 후 오른 쪽 노드로 이동하는 것을 반복하여 중위순회로 선수들의 명단과 정보들을 출력한다.

Stage3. Search the Best Team & Delete Players of Best team from the fwBST, mfBST, dfBST, and gkBST

구단주에게 받은 예산으로 구성할 수 있는 가장 강력한 팀을 만들어서 출력한다. 그리고 구성한 팀에 있는 선수들을 Binary Search Tree에서 제거하고 다시 한번 선수 명단과 정보들을 출력한다. 이를 구현하기 위해서는 노드를 삭제해줄 deletion 함수와 best team을 탐색하여 구성해줄 GetBestTeam 함수를 만들어야 한다.

- **Deletion 함수 pseudo code**

```
Void deletion(ability)
        TreeNode * Current = root node, *Prev = NULL;
        For(until do same ability with deletion ability)
                If deletion ability < current ability, move at left
                Else ability > current ability, move at right}
        If current node is null, function end
        If current->left = null and current->right null
                If prev = null, current = root
                Else if prev->right=current, prev->right delete
                Else prev->left delete
                Program ending
        If current->left = null
                If prev = null, current->right = root
                Else if prev->right=current, prev->right = current->right
                Else prev->left=current->right and program ending}
        If current->right = null
                If prev = null, current->left = root
                Else if prev->right=current, prev->right = current->left
                Else prev->left=current->left and program ending}
        TreeNode * 2prev = current, *prev=current->left
        TreeNode * cur = current->left->right
        For(cur != NULL)
                2prev=prev, prev = cur, cur->right
        Current->set data of p
        If(2prev=current), 2prev->left=prev->right
        Else, 2prev->right=prev->right
```

Deletion 함수는 ability를 인자로 받아서 tree안에서 인자로 받은 ability값과 동일한 값을 가지고 있는 노드의 위치를 current node에 저장하고, 4가지의 경우로 나누어서 구현한다.

1. child node가 없는 경우: prev, current node를 이용해 prev node가 null이면 current node 가 root 임으로 root = null, prev노드의 right node가 current node면 prev->right = 0, left node가 current node면 prev-> left = 0
2. left child node만 없는 경우: prev, current node를 이용해 prev node가 null이면 current node 가 root 임으로 root=current->right, prev->right = current면 prev->right=current->right, 아니면 prev->left=current->right
3. right child node만 없는 경우: prev, current node를 이용해 prev node가 null이면 current node 가 root 임으로 root=current->left, prev->right = current면 prev->right=current->left, 아니면 prev->left=current->left
4. child node가 둘 다 있는 경우: 이때는 current node를 지우고 왼쪽에 있는 sub tree에서 가장 값이 큰 node로 current node를 대체 해준다.

- **GetBestTeam 함수 pseudo code**

PutDataInorder(Visit, vector v)

If visit is null, return null

PutDataInorder(visit->left, v)

Push data at vector

PutDataInorder(visit->right v)

getBestTeam()

SoccerTeam best_team;

Vector fw,mf,df,gk

TreeNode * fw,mf,df,gk=m_root

fw,mf,df,gkBST.PutDataInorder(fw,mf,df,gkBST.m_root, Vec_fw,mf,df,gk)

SoccerPlayerData forward,midfielder,defender,goalkeeper

Four overlapped for loop to make a team for each forward, midfielder, defender and goalkeeper in budget(by comparison and replacement)

Fw,mf,df,gkBST.deletion(best_team_ability)

getBestTeam함수는 우선 4개 포지션의 Binary Search Tree의 데이터들을 inorder로 불러와야 한다. BinarySearchTree.h에 Vector에 Inorder로 Data를 입력해 줄 PutDataInput함수를 구현한다. 그리고 4개의 반복문을 이용하여 forward, midfielder, defender, goalkeeper 한 명씩 조합되는 모든 경우의 수를 탐색하고 주어진 예산 안에서 ability가 가장 크면서 budget이 가장 작을 때 best team의 data를 갱신한다. 그리고 마지막으로 뽑힌 선수들의 데이터를 각 BST에서 삭제한다.

3. 결과(동작 화면)

1st 4400억

```
minhyeok@ubuntu:~/ds$ ./run ShootForLog.txt 4400
********Forward List********
(node.m_name: Ronaldo), (node.m_position: Forward), (node.m_transfer_fee: 250), (node.m_ability: 63)
(node.m_name: Sterling), (node.m_position: Forward), (node.m_transfer_fee: 675), (node.m_ability: 85)
(node.m_name: Mbappe), (node.m_position: Forward), (node.m_transfer_fee: 2475), (node.m_ability: 87)
(node.m_name: Griezmann), (node.m_position: Forward), (node.m_transfer_fee: 809), (node.m_ability: 88)
(node.m_name: Kane), (node.m_position: Forward), (node.m_transfer_fee: 844), (node.m_ability: 89)
(node.m_name: Neymar), (node.m_position: Forward), (node.m_transfer_fee: 1119), (node.m_ability: 90)
(node.m_name: Lewandowski), (node.m_position: Forward), (node.m_transfer_fee: 950), (node.m_ability: 91)
(node.m_name: Salah), (node.m_position: Forward), (node.m_transfer_fee: 960), (node.m_ability: 92)
(node.m_name: Hazard), (node.m_position: Forward), (node.m_transfer_fee: 677), (node.m_ability: 93)
(node.m_name: Dybala), (node.m_position: Forward), (node.m_transfer_fee: 870), (node.m_ability: 94)
(node.m_name: Suarez), (node.m_position: Forward), (node.m_transfer_fee: 1250), (node.m_ability: 95)
(node.m_name: Seung-woo), (node.m_position: Forward), (node.m_transfer_fee: 800), (node.m_ability: 96)
(node.m_name: Jisung), (node.m_position: Forward), (node.m_transfer_fee: 1000), (node.m_ability: 97)
(node.m_name: Son), (node.m_position: Forward), (node.m_transfer_fee: 1200), (node.m_ability: 98)
(node.m_name: Messi), (node.m_position: Forward), (node.m_transfer_fee: 1300), (node.m_ability: 99)
**************************
********Midflder List********
(node.m_name: Vidal), (node.m_position: Midfielder), (node.m_transfer_fee: 389), (node.m_ability: 75)
(node.m_name: Robben), (node.m_position: Midfielder), (node.m_transfer_fee: 589), (node.m_ability: 85)
(node.m_name: Fabregas), (node.m_position: Midfielder), (node.m_transfer_fee: 721), (node.m_ability: 86)
(node.m_name: Toure), (node.m_position: Midfielder), (node.m_transfer_fee: 1011), (node.m_ability: 87)
(node.m_name: Eriksen), (node.m_position: Midfielder), (node.m_transfer_fee: 947), (node.m_ability: 88)
(node.m_name: Schweinsteiger), (node.m_position: Midfielder), (node.m_transfer_fee: 713), (node.m_ability: 91)
(node.m_name: Silva), (node.m_position: Midfielder), (node.m_transfer_fee: 342), (node.m_ability: 92)
(node.m_name: Modric), (node.m_position: Midfielder), (node.m_transfer_fee: 867), (node.m_ability: 93)
(node.m_name: Kroos), (node.m_position: Midfielder), (node.m_transfer_fee: 762), (node.m_ability: 94)
(node.m_name: Busquets), (node.m_position: Midfielder), (node.m_transfer_fee: 321), (node.m_ability: 95)
(node.m_name: Zidane), (node.m_position: Midfielder), (node.m_transfer_fee: 1115), (node.m_ability: 96)
(node.m_name: Iniesta), (node.m_position: Midfielder), (node.m_transfer_fee: 845), (node.m_ability: 97)
(node.m_name: Pogba), (node.m_position: Midfielder), (node.m_transfer_fee: 2581), (node.m_ability: 98)
(node.m_name: Kang in), (node.m_position: Midfielder), (node.m_transfer_fee: 900), (node.m_ability: 99)
**************************
********Defender List********
(node.m_name: Luiz), (node.m_position: Defender), (node.m_transfer_fee: 741), (node.m_ability: 65)
(node.m_name: Maicon), (node.m_position: Defender), (node.m_transfer_fee: 777), (node.m_ability: 77)
(node.m_name: Neville), (node.m_position: Defender), (node.m_transfer_fee: 1212), (node.m_ability: 80)
(node.m_name: Pique), (node.m_position: Defender), (node.m_transfer_fee: 813), (node.m_ability: 81)
(node.m_name: Zanetti), (node.m_position: Defender), (node.m_transfer_fee: 819), (node.m_ability: 85)
(node.m_name: Lahm), (node.m_position: Defender), (node.m_transfer_fee: 817), (node.m_ability: 88)
(node.m_name: Carlos), (node.m_position: Defender), (node.m_transfer_fee: 999), (node.m_ability: 89)
(node.m_name: Vidic), (node.m_position: Defender), (node.m_transfer_fee: 349), (node.m_ability: 92)
(node.m_name: Ramos), (node.m_position: Defender), (node.m_transfer_fee: 913), (node.m_ability: 93)
(node.m_name: Maldini), (node.m_position: Defender), (node.m_transfer_fee: 498), (node.m_ability: 94)
(node.m_name: Yeonggwon), (node.m_position: Defender), (node.m_transfer_fee: 1218), (node.m_ability: 95)
(node.m_name: Nesta), (node.m_position: Defender), (node.m_transfer_fee: 1050), (node.m_ability: 96)
(node.m_name: Puyol), (node.m_position: Defender), (node.m_transfer_fee: 924), (node.m_ability: 97)
(node.m_name: Alves), (node.m_position: Defender), (node.m_transfer_fee: 247), (node.m_ability: 98)
(node.m_name: van Dijk), (node.m_position: Defender), (node.m_transfer_fee: 1450), (node.m_ability: 99)
********Goalkeeper List********
(node.m_name: Jeong Sung-Ryong), (node.m_position: Goalkeeper), (node.m_transfer_fee: 846), (node.m_ability: 5
4)
******************************

Best Players
(node.m_name: Messi), (node.m_position: Forward), (node.m_transfer_fee: 1300), (node.m_ability: 99)
(node.m_name: Kang in), (node.m_position: Midfielder), (node.m_transfer_fee: 900), (node.m_ability: 99)
(node.m_name: Alves), (node.m_position: Defender), (node.m_transfer_fee: 247), (node.m_ability: 98)
(node.m_name: Jeong Sung-Ryong), (node.m_position: Goalkeeper), (node.m_transfer_fee: 846), (node.m_ability: 5
4)
sum_transfer_fee 3293
sum_ability 350

-----
The Transfer window close
********Forward List********
(node.m_name: Ronaldo), (node.m_position: Forward), (node.m_transfer_fee: 250), (node.m_ability: 63)
(node.m_name: Sterling), (node.m_position: Forward), (node.m_transfer_fee: 675), (node.m_ability: 85)
(node.m_name: Mbappe), (node.m_position: Forward), (node.m_transfer_fee: 2475), (node.m_ability: 87)
(node.m_name: Griezmann), (node.m_position: Forward), (node.m_transfer_fee: 809), (node.m_ability: 88)
(node.m_name: Kane), (node.m_position: Forward), (node.m_transfer_fee: 844), (node.m_ability: 89)
(node.m_name: Neymar), (node.m_position: Forward), (node.m_transfer_fee: 1119), (node.m_ability: 90)
(node.m_name: Lewandowski), (node.m_position: Forward), (node.m_transfer_fee: 950), (node.m_ability: 91)
(node.m_name: Salah), (node.m_position: Forward), (node.m_transfer_fee: 960), (node.m_ability: 92)
(node.m_name: Hazard), (node.m_position: Forward), (node.m_transfer_fee: 677), (node.m_ability: 93)
(node.m_name: Dybala), (node.m_position: Forward), (node.m_transfer_fee: 870), (node.m_ability: 94)
(node.m_name: Suarez), (node.m_position: Forward), (node.m_transfer_fee: 1250), (node.m_ability: 95)
(node.m_name: Seung-woo), (node.m_position: Forward), (node.m_transfer_fee: 800), (node.m_ability: 96)
(node.m_name: Jisung), (node.m_position: Forward), (node.m_transfer_fee: 1000), (node.m_ability: 97)
(node.m_name: Son), (node.m_position: Forward), (node.m_transfer_fee: 1200), (node.m_ability: 98)
**************************
********Midflder List********
(node.m_name: Vidal), (node.m_position: Midfielder), (node.m_transfer_fee: 389), (node.m_ability: 75)
(node.m_name: Robben), (node.m_position: Midfielder), (node.m_transfer_fee: 589), (node.m_ability: 85)
(node.m_name: Fabregas), (node.m_position: Midfielder), (node.m_transfer_fee: 721), (node.m_ability: 86)
(node.m_name: Toure), (node.m_position: Midfielder), (node.m_transfer_fee: 1011), (node.m_ability: 87)
(node.m_name: Eriksen), (node.m_position: Midfielder), (node.m_transfer_fee: 947), (node.m_ability: 88)
(node.m_name: Schweinsteiger), (node.m_position: Midfielder), (node.m_transfer_fee: 713), (node.m_ability: 91)
(node.m_name: Silva), (node.m_position: Midfielder), (node.m_transfer_fee: 342), (node.m_ability: 92)
(node.m_name: Modric), (node.m_position: Midfielder), (node.m_transfer_fee: 867), (node.m_ability: 93)
(node.m_name: Kroos), (node.m_position: Midfielder), (node.m_transfer_fee: 762), (node.m_ability: 94)
(node.m_name: Busquets), (node.m_position: Midfielder), (node.m_transfer_fee: 321), (node.m_ability: 95)
(node.m_name: Zidane), (node.m_position: Midfielder), (node.m_transfer_fee: 1115), (node.m_ability: 96)
(node.m_name: Iniesta), (node.m_position: Midfielder), (node.m_transfer_fee: 845), (node.m_ability: 97)
(node.m_name: Pogba), (node.m_position: Midfielder), (node.m_transfer_fee: 2581), (node.m_ability: 98)
**************************
********Defender List********
(node.m_name: Luiz), (node.m_position: Defender), (node.m_transfer_fee: 741), (node.m_ability: 65)
```

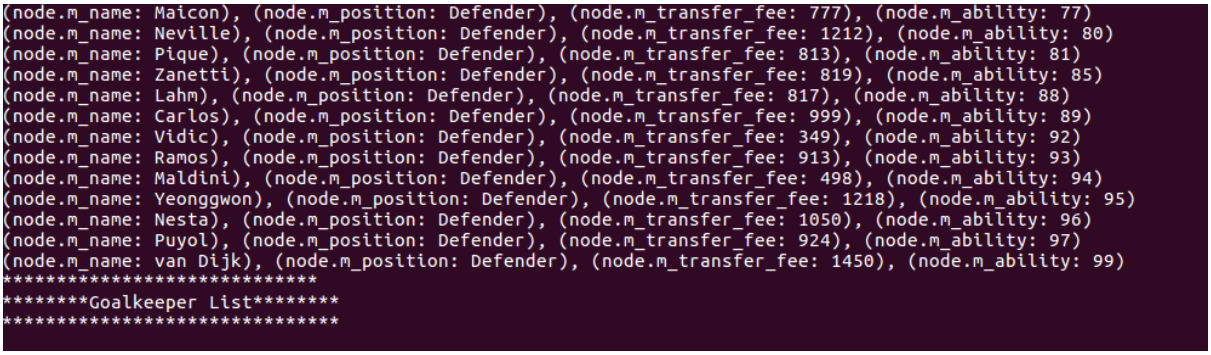
```
(node.m_name: Maicon), (node.m_position: Defender), (node.m_transfer_fee: 777), (node.m_ability: 77)
(node.m_name: Neville), (node.m_position: Defender), (node.m_transfer_fee: 1212), (node.m_ability: 80)
(node.m_name: Pique), (node.m_position: Defender), (node.m_transfer_fee: 813), (node.m_ability: 81)
(node.m_name: Zanetti), (node.m_position: Defender), (node.m_transfer_fee: 819), (node.m_ability: 85)
(node.m_name: Lahm), (node.m_position: Defender), (node.m_transfer_fee: 817), (node.m_ability: 88)
(node.m_name: Carlos), (node.m_position: Defender), (node.m_transfer_fee: 999), (node.m_ability: 89)
(node.m_name: Vidic), (node.m_position: Defender), (node.m_transfer_fee: 349), (node.m_ability: 92)
(node.m_name: Ramos), (node.m_position: Defender), (node.m_transfer_fee: 913), (node.m_ability: 93)
(node.m_name: Maldini), (node.m_position: Defender), (node.m_transfer_fee: 498), (node.m_ability: 94)
(node.m_name: Yeonggwon), (node.m_position: Defender), (node.m_transfer_fee: 1218), (node.m_ability: 95)
(node.m_name: Nesta), (node.m_position: Defender), (node.m_transfer_fee: 1050), (node.m_ability: 96)
(node.m_name: Puyol), (node.m_position: Defender), (node.m_transfer_fee: 924), (node.m_ability: 97)
(node.m_name: van Dijk), (node.m_position: Defender), (node.m_transfer_fee: 1450), (node.m_ability: 99)
*****************************
********Goalkeeper List********
*****************************
```

2nd 2000억

```
mtnhyeok@ubuntu:~/ds$ ./run ShootForLog.txt 2000
********Forward List********
(node.m_name: Ronaldo), (node.m_position: Forward), (node.m_transfer_fee: 250), (node.m_ability: 63)
(node.m_name: Sterling), (node.m_position: Forward), (node.m_transfer_fee: 675), (node.m_ability: 85)
(node.m_name: Mbappe), (node.m_position: Forward), (node.m_transfer_fee: 2475), (node.m_ability: 87)
(node.m_name: Griezmann), (node.m_position: Forward), (node.m_transfer_fee: 809), (node.m_ability: 88)
(node.m_name: Kane), (node.m_position: Forward), (node.m_transfer_fee: 844), (node.m_ability: 89)
(node.m_name: Neymar), (node.m_position: Forward), (node.m_transfer_fee: 1119), (node.m_ability: 90)
(node.m_name: Lewandowski), (node.m_position: Forward), (node.m_transfer_fee: 950), (node.m_ability: 91)
(node.m_name: Salah), (node.m_position: Forward), (node.m_transfer_fee: 960), (node.m_ability: 92)
(node.m_name: Hazard), (node.m_position: Forward), (node.m_transfer_fee: 677), (node.m_ability: 93)
(node.m_name: Dybala), (node.m_position: Forward), (node.m_transfer_fee: 870), (node.m_ability: 94)
(node.m_name: Suarez), (node.m_position: Forward), (node.m_transfer_fee: 1250), (node.m_ability: 95)
(node.m_name: Seung-woo), (node.m_position: Forward), (node.m_transfer_fee: 800), (node.m_ability: 96)
(node.m_name: Jisung), (node.m_position: Forward), (node.m_transfer_fee: 1000), (node.m_ability: 97)
(node.m_name: Son), (node.m_position: Forward), (node.m_transfer_fee: 1200), (node.m_ability: 98)
(node.m_name: Messi), (node.m_position: Forward), (node.m_transfer_fee: 1300), (node.m_ability: 99)
*****************************
********Midflder List********
(node.m_name: Vidal), (node.m_position: Midfielder), (node.m_transfer_fee: 389), (node.m_ability: 75)
(node.m_name: Robben), (node.m_position: Midfielder), (node.m_transfer_fee: 589), (node.m_ability: 85)
(node.m_name: Fabregas), (node.m_position: Midfielder), (node.m_transfer_fee: 721), (node.m_ability: 86)
(node.m_name: Toure), (node.m_position: Midfielder), (node.m_transfer_fee: 1011), (node.m_ability: 87)
(node.m_name: Eriksen), (node.m_position: Midfielder), (node.m_transfer_fee: 947), (node.m_ability: 88)
(node.m_name: Schweinsteiger), (node.m_position: Midfielder), (node.m_transfer_fee: 713), (node.m_ability: 91)
(node.m_name: Silva), (node.m_position: Midfielder), (node.m_transfer_fee: 342), (node.m_ability: 92)
(node.m_name: Modric), (node.m_position: Midfielder), (node.m_transfer_fee: 867), (node.m_ability: 93)
(node.m_name: Kroos), (node.m_position: Midfielder), (node.m_transfer_fee: 762), (node.m_ability: 94)
(node.m_name: Busquets), (node.m_position: Midfielder), (node.m_transfer_fee: 321), (node.m_ability: 95)
(node.m_name: Zidane), (node.m_position: Midfielder), (node.m_transfer_fee: 1115), (node.m_ability: 96)
(node.m_name: Iniesta), (node.m_position: Midfielder), (node.m_transfer_fee: 845), (node.m_ability: 97)
(node.m_name: Pogba), (node.m_position: Midfielder), (node.m_transfer_fee: 2581), (node.m_ability: 98)
(node.m_name: Kang in), (node.m_position: Midfielder), (node.m_transfer_fee: 900), (node.m_ability: 99)
*****************************
********Defender List********
(node.m_name: Luiz), (node.m_position: Defender), (node.m_transfer_fee: 741), (node.m_ability: 65)
(node.m_name: Maicon), (node.m_position: Defender), (node.m_transfer_fee: 777), (node.m_ability: 77)
(node.m_name: Neville), (node.m_position: Defender), (node.m_transfer_fee: 1212), (node.m_ability: 80)
(node.m_name: Pique), (node.m_position: Defender), (node.m_transfer_fee: 813), (node.m_ability: 81)
(node.m_name: Zanetti), (node.m_position: Defender), (node.m_transfer_fee: 819), (node.m_ability: 85)
(node.m_name: Lahm), (node.m_position: Defender), (node.m_transfer_fee: 817), (node.m_ability: 88)
(node.m_name: Carlos), (node.m_position: Defender), (node.m_transfer_fee: 999), (node.m_ability: 89)
(node.m_name: Vidic), (node.m_position: Defender), (node.m_transfer_fee: 349), (node.m_ability: 92)
(node.m_name: Ramos), (node.m_position: Defender), (node.m_transfer_fee: 913), (node.m_ability: 93)
(node.m_name: Maldini), (node.m_position: Defender), (node.m_transfer_fee: 498), (node.m_ability: 94)
(node.m_name: Yeonggwon), (node.m_position: Defender), (node.m_transfer_fee: 1218), (node.m_ability: 95)
(node.m_name: Nesta), (node.m_position: Defender), (node.m_transfer_fee: 1050), (node.m_ability: 96)
(node.m_name: Puyol), (node.m_position: Defender), (node.m_transfer_fee: 924), (node.m_ability: 97)
(node.m_name: Alves), (node.m_position: Defender), (node.m_transfer_fee: 247), (node.m_ability: 98)
(node.m_name: van Dijk), (node.m_position: Defender), (node.m_transfer_fee: 1450), (node.m_ability: 99)
*****************************
********Goalkeeper List********
(node.m_name: Jeong Sung-Ryong), (node.m_position: Goalkeeper), (node.m_transfer_fee: 846), (node.m_ability: 54)
*****************************

Best Players
(node.m_name: Ronaldo), (node.m_position: Forward), (node.m_transfer_fee: 250), (node.m_ability: 63)
(node.m_name: Busquets), (node.m_position: Midfielder), (node.m_transfer_fee: 321), (node.m_ability: 95)
(node.m_name: Alves), (node.m_position: Defender), (node.m_transfer_fee: 247), (node.m_ability: 98)
(node.m_name: Jeong Sung-Ryong), (node.m_position: Goalkeeper), (node.m_transfer_fee: 846), (node.m_ability: 54)
sum_transfer_fee 1664
sum_ability 310

-----
The Transfer window close
********Forward List********
(node.m_name: Sterling), (node.m_position: Forward), (node.m_transfer_fee: 675), (node.m_ability: 85)
(node.m_name: Mbappe), (node.m_position: Forward), (node.m_transfer_fee: 2475), (node.m_ability: 87)
(node.m_name: Griezmann), (node.m_position: Forward), (node.m_transfer_fee: 809), (node.m_ability: 88)
(node.m_name: Kane), (node.m_position: Forward), (node.m_transfer_fee: 844), (node.m_ability: 89)
(node.m_name: Neymar), (node.m_position: Forward), (node.m_transfer_fee: 1119), (node.m_ability: 90)
(node.m_name: Lewandowski), (node.m_position: Forward), (node.m_transfer_fee: 950), (node.m_ability: 91)
(node.m_name: Salah), (node.m_position: Forward), (node.m_transfer_fee: 960), (node.m_ability: 92)
(node.m_name: Hazard), (node.m_position: Forward), (node.m_transfer_fee: 677), (node.m_ability: 93)
(node.m_name: Dybala), (node.m_position: Forward), (node.m_transfer_fee: 870), (node.m_ability: 94)
(node.m_name: Suarez), (node.m_position: Forward), (node.m_transfer_fee: 1250), (node.m_ability: 95)
(node.m_name: Seung-woo), (node.m_position: Forward), (node.m_transfer_fee: 800), (node.m_ability: 96)
(node.m_name: Jisung), (node.m_position: Forward), (node.m_transfer_fee: 1000), (node.m_ability: 97)
(node.m_name: Son), (node.m_position: Forward), (node.m_transfer_fee: 1200), (node.m_ability: 98)
(node.m_name: Messi), (node.m_position: Forward), (node.m_transfer_fee: 1300), (node.m_ability: 99)
*****************************
********Midflder List********
(node.m_name: Vidal), (node.m_position: Midfielder), (node.m_transfer_fee: 389), (node.m_ability: 75)
(node.m_name: Robben), (node.m_position: Midfielder), (node.m_transfer_fee: 589), (node.m_ability: 85)
(node.m_name: Fabregas), (node.m_position: Midfielder), (node.m_transfer_fee: 721), (node.m_ability: 86)
(node.m_name: Toure), (node.m_position: Midfielder), (node.m_transfer_fee: 1011), (node.m_ability: 87)
(node.m_name: Eriksen), (node.m_position: Midfielder), (node.m_transfer_fee: 947), (node.m_ability: 88)
(node.m_name: Schweinsteiger), (node.m_position: Midfielder), (node.m_transfer_fee: 713), (node.m_ability: 91)
(node.m_name: Silva), (node.m_position: Midfielder), (node.m_transfer_fee: 342), (node.m_ability: 92)
(node.m_name: Modric), (node.m_position: Midfielder), (node.m_transfer_fee: 867), (node.m_ability: 93)
(node.m_name: Kroos), (node.m_position: Midfielder), (node.m_transfer_fee: 762), (node.m_ability: 94)
(node.m_name: Zidane), (node.m_position: Midfielder), (node.m_transfer_fee: 1115), (node.m_ability: 96)
(node.m_name: Iniesta), (node.m_position: Midfielder), (node.m_transfer_fee: 845), (node.m_ability: 97)
(node.m_name: Pogba), (node.m_position: Midfielder), (node.m_transfer_fee: 2581), (node.m_ability: 98)
(node.m_name: Kang in), (node.m_position: Midfielder), (node.m_transfer_fee: 900), (node.m_ability: 99)
*****************************
********Defender List********
(node.m_name: Luiz), (node.m_position: Defender), (node.m_transfer_fee: 741), (node.m_ability: 65)
```



4. 고찰

보고서를 쓰면서 의사(pseudo) 코드에 대해 공부하게 되었다. 평소 강의를 수강하면서 보기만 했지 직접 작성해 보는 것은 처음이라 감이 잡히지 않았다. 표준화되지 않았지만 포트란 스타일, 파스칼 스타일, C 스타일 등 여러 스타일이 있다는 것에 대해 알게 되었다. 이를 참고하여 표준화되지 않았고, 각각 다른 언어들을 사용하는 개발자들이 서로의 코드를 알아보기 쉽게 작성한다는 특성에 중점을 두어 작성해보았다.

이번 과제를 하면서 Binary Search Tree에 대한 깊은 공부를 할 수 있었다. 과제의 핵심은 node를 insert하고 delete하고 inorder로 순회하는 기능을 구현하는 것이었다. 구현하면서 계속해서 Tree 구조를 머릿속에서 그리기도 하고 직접 종이에 그려 보기도 하면서 코드 한 줄 한 줄 깊게 다가갈 수 있었던 기회였다. 특히 delete의 경우 4가지로 나누어서 생각했어야 했는데 child node 2개 모두를 가지고 있을 때가 가장 어려웠다. 데이터 구조 설계 강의자료에서 많은 도움을 받았는데 3개의 node prevprev, prev, current를 함께 움직이면서 prev가 삭제 할 노드가 된다는 점을 이해하기까지 많은 시간이 걸렸다.

그리고 member들을 불러와 불필요한 변수 선언을 줄임으로써 반복적인 코드들을 줄여주는 class, struct와 문자열을 한 덩어리 씩 저장할 수 있는vector, 문자열 데이터 들을 효율적으로 다룰 수 있는 string, 데이터를 Last input first output 방식으로 저장하고 내보내는 stack에 대해 복습할 수 있는 좋은 기회가 되었다.

마지막으로 요즘 대부분의 사람들이 쓰는 windows가 아닌 linux상에서

Code를 compile하고 실행시키는 방식으로 과제를 진행하게 되었는데,

처음으로 다른 운영체제를 다뤄보면서 운영체제마다 장단점이 있다는 것을 깨닫게 되었다.