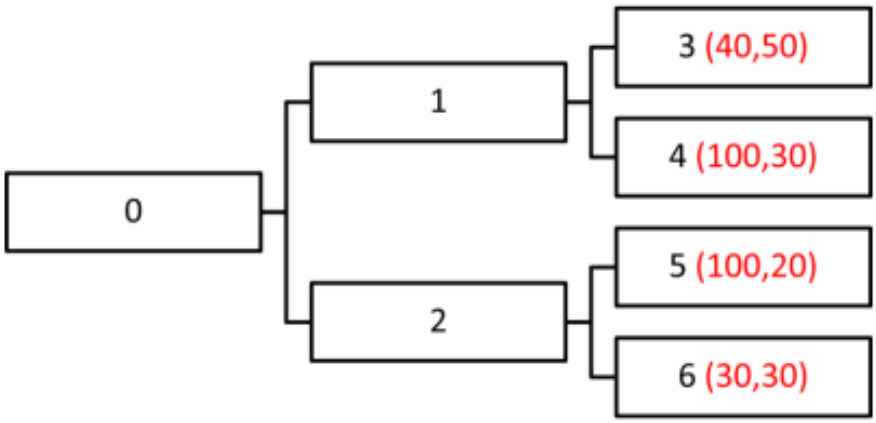
MCP361 : Industrial Engineering Lab: Assignment 8

Protocol to represent the games



Suppose we have a tree as given above. This game can be represented as a Breadth first traversal.
So we will write its Breadth first traversal as follows-
0: ;2: ;1: ;6:30,30;5:100,20;4:100,30;3:40,50
Each node is separated by a semicolon(;), each node starts with a node name, then a colon(:) and then its payoff vector. Initially all the non-leaf nodes will have None as a payoff vector and all the leaf nodes have a payoff vector.

Protocol for reading the games from text file

```
l = []
with open('1.txt') as f:
        l = f.readlines()
```

From the above lines we get a l array as follows-
['0: ;2: ;1: ;6:30,30;5:100,20;4:100,30;3:40,50']

```
d=[]

for line in lines:
d.append(line.replace("\n",""))
        d = d[0].split(";")
for i in range(len(d)):
        d[i] = d[i].split(":")
```

Form the above lines we get a d array as follows-
[['0', ' '], ['2', ' '], ['1', ' '], ['6', '30,30'], ['5', '100,20'], ['4', '100,30'], ['3', '40,50']]

```
f = []
for i in range(len(d)):
        temp = []
        n = int(d[i][0])
        temp.append(n)
        if d[i][1]==" ":
                temp.append(None)
```

```
        else:
                val = list(map(int, d[i][1].split(",")))
                temp.append(val)
        f.append(temp)
```

Form the above lines we get a f array as follows-
[[0, None], [2, None], [1, None], [6, [30, 30]], [5, [100, 20]], [4, [100, 30]], [3, [40, 50]]]

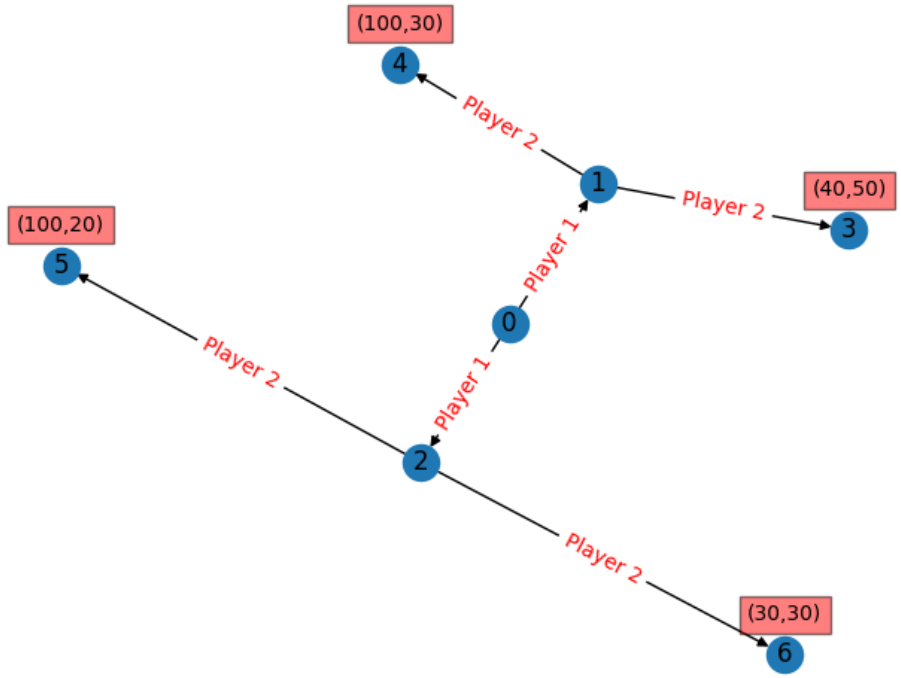
Now, with this array f we will form a tree whose nodes will be the same as the elements in array f. Array f contains node information in Breadth first traversal order so it is easy to form a tree if we know its breadth first traversal order. I have implemented a function named createTree which is used to create a tree.

The tree class is shown belowclass
TreeNode:

```
def __init__(self, n, x):
        self.val = x
        self.left = None
        self.right = None
        self.node = n
```

Output & solution for Problem1:

```
= RESTART: E:\Semester 7\MCP361\MCP361_2018ME20727_Assignment8\MCP361_2018ME20727_Assignment8.py
Solution for Problem 1
At node 0 Player 1 chooses to move to node 1
At node 1 Player 2 chooses to move to node 3
The backward induction strategy ends at node 3
The optimal payoff vector is [40, 50]
```



Output & Solution for Problem 2:

```
Solution For Problem 2
At node 0 Player 1 chooses to move to node 1
At node 1 Player 2 chooses to move to node 4
The backward induction strategy ends at node 4
The optimal payoff vector is [160, 0]
```

Output & Solution for Problem 3:

```
Solution For Problem 3
At node 0 Player 1 chooses to move to node 1
At node 1 Player 2 chooses to move to node 4
At node 4 Player 3 chooses to move to node 10
The backward induction strategy ends at node 10
The optimal payoff vector is [50, 20, 20]
```