

Exercise 6 – Minimax Algorithm

Date: 06/10/2022

Aim:

To implement the minimax algorithm by doing the following:

- Consider a state space where the start state is number '1' and its successor function returns $2n+1$ and $2n+2$. Construct a state space tree (Full binary tree) for the level specified by the user.
- Write and Implement the Minimax algorithm as a Recursive Depth First Search using minimax function. Assume first move is carried out by max player and last level is terminal nodes.

Code:

```
import math
from binarytree import Node
from binarytree import build

def left(seq,index):
    return seq[2*index+1]

def right(seq,index):
    return seq[2*index+2]

level = int(input("\nEnter level limit: "))
ht = level+1

seq = [i for i in range(1,2**ht)]
tree = build(seq)
print(tree)
nodes = tree.values
print("Nodes:",nodes)

# Recursive DFS for Minimax algorithm

value = None
for l in range(level-1,-1,-1):
```

```
num = 2**l
for i in range(num-1,num*2-1):
    if (l%2 == 0):
        value = max(left(nodes,i), right(nodes,i))
    else:
        value = min(left(nodes,i), right(nodes,i))
    nodes[i] = value
```

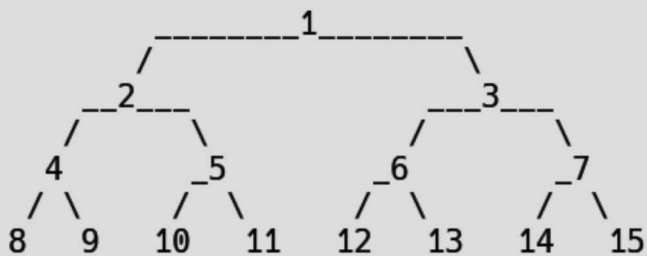
```
answerTree = build(nodes)
print(answerTree)
```

```
print("Answer:", value)
```

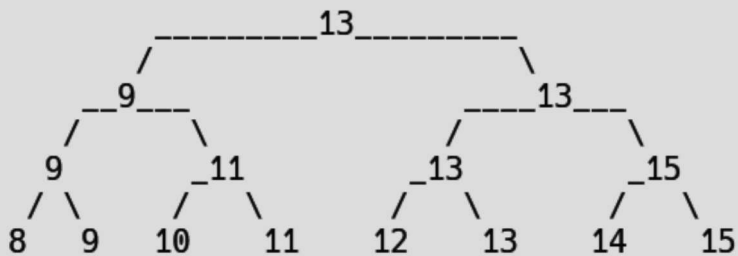
Output:

```
~/AIwork$ python3 minmax.py
```

```
Enter level limit: 3
```



```
Nodes: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```



```
Answer: 13
```

```
~/AIwork$
```