AI Assignment - 1
Breadth First Search and Depth First Search

Sabarivasan Velayutham
205001085
CSE-B

Algorithm:

1.Get input of max value of n
2.Get color input either red or green
3.Depending on the color generate arrays
4.Generate tree using dictionary
5. Use BFS and DFS algo on the generated tree and print the output

BFS:

Uses Queue(FIFO)

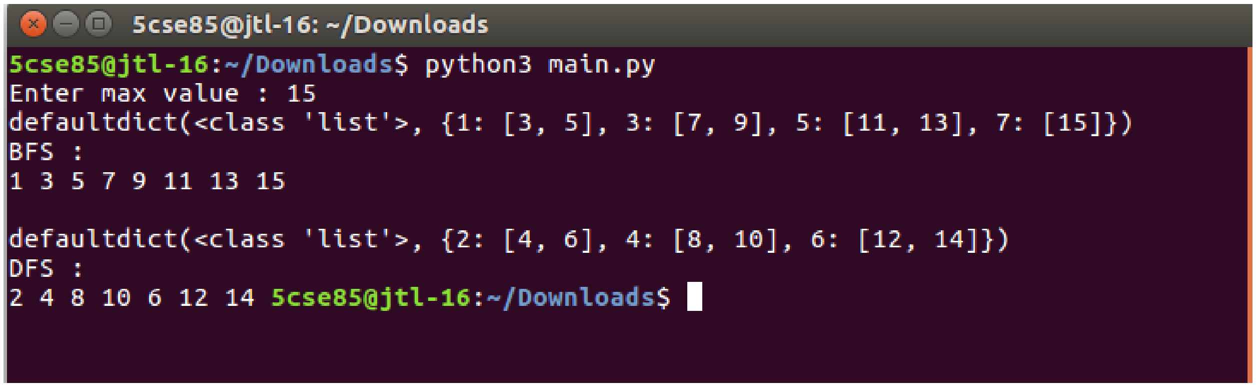
DFS:

Uses Stack(LIFO)

Code :

```
from collections import deque, defaultdict


def redballs(n):
    red_arr = [i for i in range(1, n+1, 2)]
    return red_arr


def greenballs(n):
    green_arr = [i for i in range(2, n+1, 2)]
    return green_arr


def bfs(graph, visited, root):

    queue = deque([root])
    visited.add(root)

    while queue:
        vertex = queue.popleft()
        print(vertex, end=" ")
        if vertex in graph.keys():
            for neighbour in graph[vertex]:
                if neighbour not in visited:
                    visited.add(neighbour)
                    queue.append(neighbour)


def dfs(graph, visited, root):
    if root not in visited:
        print(root, end=' ')
        visited.add(root)
        if root in graph.keys():
            for neighbour in graph[root]:
                dfs(graph, visited, neighbour)


# graph = defaultdict(list)
# graph = {0: [1, 2], 1: [3,4], 2: [5,6]}
# visited = set()
# dfs(graph,visited, 0)

graph1 = defaultdict(list)
graph2 = defaultdict(list)

n = int(input("Enter max value : "))
rballs = redballs(n)
gballs = greenballs(n)

visited1 = set()
visited2 = set()

for i in range(len(rballs)//2):
    try:
        graph1[rballs[i]] = [rballs[2*i+1], rballs[2*i+2]]
    except:
        graph1[rballs[i]] = [rballs[2*i+1]]

print(graph1)
print("BFS : ")
bfs(graph1, visited1, 1)

print('\n')

for i in range(len(gballs)//2):
    try:
        graph2[gballs[i]] = [gballs[2*i+1], gballs[2*i+2]]
    except:
        graph2[gballs[i]] = [gballs[2*i+1]]

print(graph2)
print("DFS : ")
dfs(graph2, visited2, 2)
```

Output :



```
5cse85@jtl-16:~/Downloads$ python3 main.py
Enter max value : 15
defaultdict(<class 'list'>, {1: [3, 5], 3: [7, 9], 5: [11, 13], 7: [15]})
BFS :
1 3 5 7 9 11 13 15

defaultdict(<class 'list'>, {2: [4, 6], 4: [8, 10], 6: [12, 14]})
DFS :
2 4 8 10 6 12 14 5cse85@jtl-16:~/Downloads$ 
```