Lab - 04

ACT : Advanced Compiler Techniques

AIM : Extract Natural Loops present in the given Control Flow Graph.

Name : Ambalia Harshit
Roll no : –
Date : 11 Sept 2023

Question 01 : Extract Natural Loops present in the given Control Flow Graph.

Input : Back Edge
Output: Nodes of the Natural loop for the given Back Edge.

```
def predecessors(arr, size):
   predecessor_dict = {}
   for i in range(size):
       for j in range(size):
           if arr[i][j] == 1:
               if j not in predecessor_dict:
                   predecessor_dict[j] = [i]
               else:
                   predecessor_dict[j].append(i)
           else:
               if i not in predecessor_dict:
                   predecessor_dict[j] = []
   return predecessor_dict

def insert(i, loop, stack):
   if i not in loop:
```

```
        loop.append(i)
        stack.append(i)

def find_loop(predecessor, n, d):
    loop = []
    stack = []
    loop.append(d)
    insert(n, loop, stack)
    while(len(stack)>0):
        tmp = stack.pop()
        for i in predecessor[tmp]:
            insert(i, loop, stack)
    return loop

if __name__=="__main__":
    size = int(input("Enter size : "))
    # arr = []
    # for i in range(size):
    #     a = []
    #     for j in range(size):
    #         a.append(int(input()))
    #     arr.append(a)
    arr = [
        [0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
        [0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
        [0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0],
        [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
        [0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1],
        [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    ]
    print("For valid back edge n->d")
    n = int(input("Enter n : "))
    d = int(input("Enter d : "))
    n -= 1
    d -= 1
```

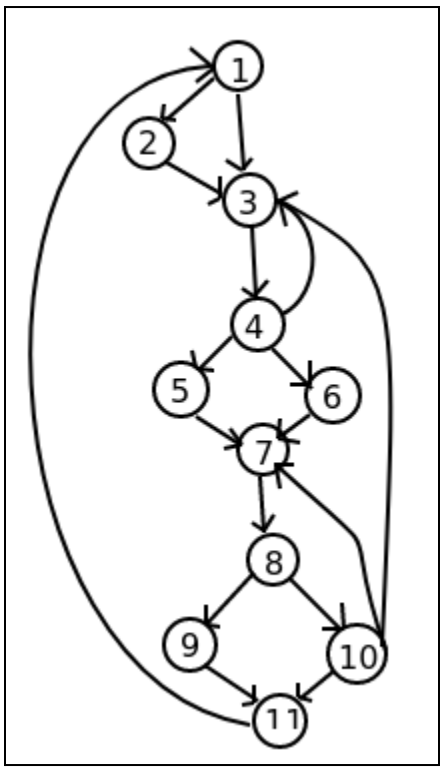
```
predecessor = predecessors(arr, size)
loop = find_loop(predecessor, n, d)

# Increases value by 1 and create distinct set
for i in range(len(loop)):
    loop[i] = loop[i]+1
distinct_list = set(loop)

print(distinct_list)
```

```
hr@Edith:~/Documents/Semeste
Lab04.py"
Enter size : 11
For valid back edge n->d
Enter n : 10
Enter d : 7
{8, 10, 7}
```

Validating the output for given graph



Back Edge	Natural loops
4 -> 3	{3, 4}
7 -> 3	{3, 4 ,5, 6 ,7 ,8, 10}
10 -> 7	{7, 8, 10}
10 -> 3	{3, 4 ,5, 6 ,7 ,8, 10}
11 -> 1	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}