Iterative Deepening depth First

```
class graph:
def __init__(self, vertex):
    self.V = vertex
    self.node_arr = []
    self.graph = defaultdict(list)

def addEdges(self, x, y):
    self.graph[x].append(y)

def DLS(self, target, depth, arr, src):
    if src == target:
        arr.append(src)
        return true

    if depth <= 0:
        return false
        arr.append(src)

    for i in self.graph[src]:
        if (self.DLS(i, target, depth-1, arr)):
            return true
            arr.pop
            return false

G = Graph(7);
```

Scanned with CamScanner

Shreyas M
1BM18CS105

```
g.addedge(0, 1)
g.addedge(0, 2)
g.addedge(1, 3)
g.addedge(1, 4)
g.addedge(2, 5)
g.addedge(2, 6)

target = 8;
depth = 6;
src = 0

if g.DFS(src, target, depth) == true:
    print("Source can reach target")
else:
    print("Source cannot reach target")

print(g.node)
```