Iterative Deepening Depth-First Search

```
class Graph:

    function __init__(vertices):
        v = vertices
        graph = {i:[] for i in range(vertices)}

    function add_edges(u,v):
            graph[u].append(v)

    function dls(src, target, limit):
        if src == target:
            return True

        if limit <=0 :
            return False

        for neighbour in graph[src]:
            if dls[neighbor, target, limit-1):
                return True

        return False

    function iddfs(src, target, max_depth):
        for depth in range(0, max_depth):
            if dls(src, target, depth):
                return True
        return False
```

Initialize graph g with 7 vertices
Add edges: (0,1), (0,2), (1,3), (1,4), (2,5), (2,

set src = 0, target = 6, max_depth = 3.

if g.iddfs(src, target, max_depth):
    print("Target found within depth")
else:
    print("Target not found within depth")

Output:

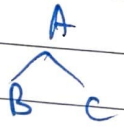
Target 6 found within depth 3.

State flow Diagram:

Depth level 0

A

Depth level 1



Depth level 2