

### Program 3: Implement Iterative deepening search algorithm.

Write\_up:

```
PRANAV JAGADEESH IBM18CS071 PROGRAM 3

from collections import defaultdict
class Graph:
    def __init__(self, vertices):
        self.V = vertices
        self.graph = defaultdict(list)
    def addEdge(self, u, v):
        self.graph[u].append(v)
    def DLS(self, src, target, maxDepth):
        if src == target: return True
        if maxDepth <= 0: return False
        for i in self.graph[src]:
            if (self.DLS(i, target, maxDepth-1)):
                return True
        return False
    def IDDFS(self, src, target, maxDepth):
        for i in range(maxDepth):
            if (self.DLS(src, target, i)):
                return True
        return False
g = Graph(7)
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 = 6; maxDepth = 3; src = 0
if g.IDDFS(src, target, maxDepth) == True:
    print("Target is reachable from source "+
          "within max depth")
else:
    print("Target is NOT reachable from source "+
          "within max depth")
```

Program:

```
from collections import defaultdict
```

```
class Graph:
```

```
    def __init__(self,vertices):
```

```
        self.V = vertices
```

```
        self.graph = defaultdict(list)
```

```
    def addEdge(self,u,v):
```

```
        self.graph[u].append(v)
```

```
    def DLS(self,src,target,maxDepth):
```

```
        if src == target : return True
```

```
        if maxDepth <= 0 : return False
```

```
        for i in self.graph[src]:
```

```
            if(self.DLS(i,target,maxDepth-1)):
```

```
                return True
```

```
        return False
```

```
    def IDDFS(self,src, target, maxDepth):
```

```

        for i in range(maxDepth):
            if (self.DLS(src, target, i)):
                return True
    return False

```

```

g = Graph (7);
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 = 6; maxDepth = 3; src = 0

```

```

if g.IDDFS(src, target, maxDepth) == True:
    print ("Target is reachable from source " +
          "within max depth")
else :
    print ("Target is NOT reachable from source " +
          "within max depth")

```

Output:

```

Target is reachable from source within max depth

```

```

Process finished with exit code 0

```