Name: Warren Fernandes Roll no: 8940

TE Comps B (Batch B)

AI Practical 3 - Uninformed Search

Iterative Deepening Search

IDDFS combines depth-first search’s space-efficiency and breadthfirst search’s fast search (for nodes closer to root).

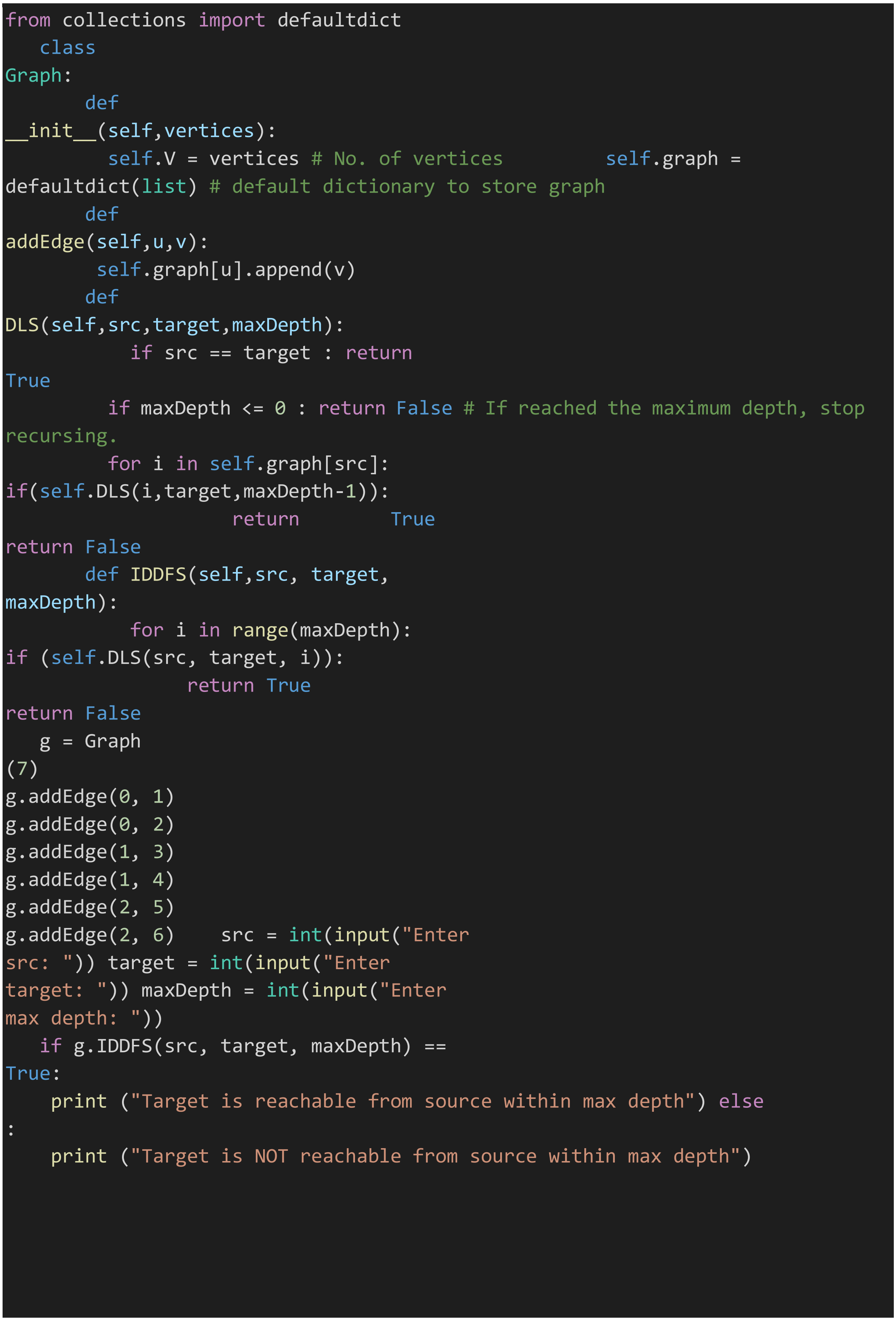
How does IDDFS work?

IDDFS calls DFS for different depths starting from an initial value. In every call, DFS is restricted from going beyond given depth. So basically, we do DFS in a BFS fashion.

Algorithm:

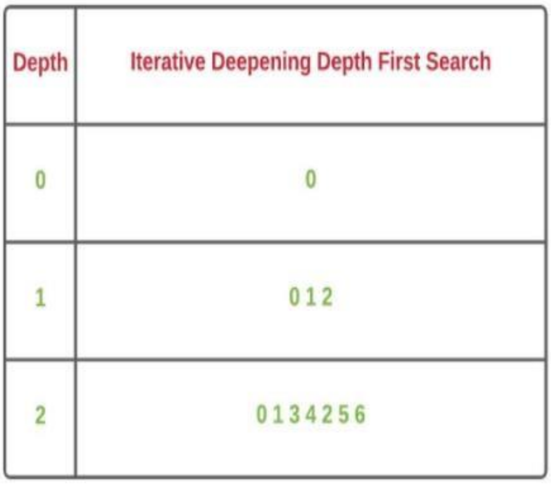
```
bool IDDFS(src, target, max_depth)
for limit from 0 to max_depth        if
DLS(src, target, limit) == true
return true     return false
 bool DLS(src, target,
limit)     if (src ==
target)         return true;

    // If reached the maximum depth,
    // stop recursing.
if   (limit   <=   0)
return false;

    foreach adjacent i of src
        if DLS(i, target, limit?1)
return true

    return false
```

Source code:

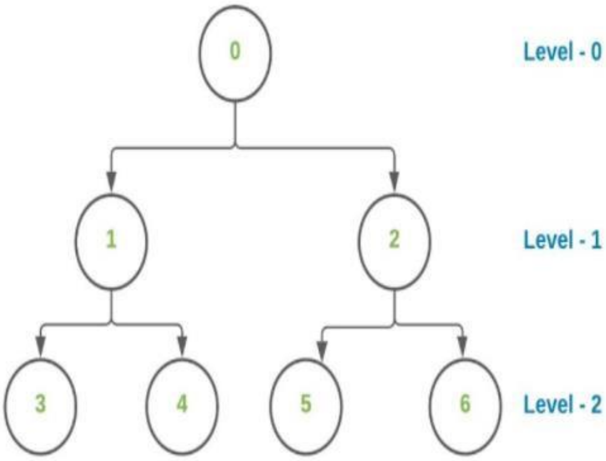


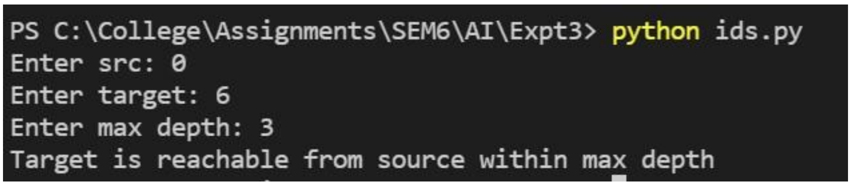
```
from collections import defaultdict
   class
Graph:
       def
__init__(self,vertices):
         self.V = vertices # No. of vertices          self.graph =
defaultdict(list) # default dictionary to store graph
       def
addEdge(self,u,v):
        self.graph[u].append(v)
       def
DLS(self,src,target,maxDepth):
           if src == target : return
True
         if maxDepth <= 0 : return False # If reached the maximum depth, stop
recursing.
         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)    src = int(input("Enter
src: ")) target = int(input("Enter
target: ")) maxDepth = int(input("Enter
max depth: "))
   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:



Depth	Iterative Deepening Depth First Search
0	0
1	0 1 2
2	0 1 3 4 2 5 6





```
PS C:\College\Assignments\SEM6\AI\Expt3> python ids.py
Enter src: 0
Enter target: 6
Enter max depth: 3
Target is reachable from source within max depth
```