Ahmed Ali
1BM18CS135

papergrid

Date: 9/11/20

AIM To find optimal path using iterative deepening search.

Algorithm: Combine depth-first search's space efficiency and breadth-first search's quick searching.

~~def [illegible]~~

```
def iterativeDFS(graph, v, discovered):
    stack = deque()
    stack.append(v)
    while stack:
        v = stack.pop()
        if discovered[v]:
            continue
        discovered[v] = True
        print(v, end='')
        adj = graph.adjList[v]
        for i in reversed(range(len(adj))):
            u = adj[i]
            if not discovered[u]:
                stack.append(u).
```

```
edges = [ as per diagram
        . - - - - - ]

N = 8
graph = Graph(edges, N)
discovered = [False] * N

for i is in range(N):
    if not discovered[i]:
        interative DFS(graph, i, discovered)
```