

Name: Soumitri Chattopadhyay

Roll: 001911001083

Department: Information Technology

Year: UG-III (5th Semester)

Subject: Artificial Intelligence Laboratory

Assignment: 3 – “Implement Iterative Deepening”

Date: 25.09.2021

Assignment Details:

Please use the following graph for the assignment. [Attached in Classroom announcement]

Things to do:

1. Implement the iterative deepening algorithm
 2. Execute the implementation on the attached graph with depth 5
 3. Execute the previously implemented DFS with depth 5 on the same graph presented here.
 4. For both points 2 and 3, the graph will be written in a file and not in the program implementation
 5. Write a document based on your observations of output 2 and the difference between the outputs of points 2 and 3.
- -----

Observations and Discussion:

1. Input file: input_AI_Ass3_83.txt

12

4

S 011100000000

p 000010000000

e 000001100000

d 000000001100

```

q 000000000000
h 010010000000
r 000000010000
f 000000000101
b 000000000010
c 000000000010
a 000000000000
G 000000000000
S
G

```

Where, the first two lines provide number of nodes and maximum depth respectively, followed by an adjacency matrix where the first letter of each row denotes vertex name, followed by a binary string such that '1' represents an edge between the respective indexed vertices and '0' represents absence of edge. The last two lines provide the source and destination vertex respectively. From the figure, it was intuitive that node 'S' and node 'G' was the longest path and so this source-destination pair has been taken for best understanding of differences between conditional DFS (Depth First Search) and IDS (Iterative Deepening Search).

2. Comparison and Discussion on Observed Outputs

<u>DFS with conditional max. depth</u>	<u>IDS (max. depth always provided)</u>
<u>Output:</u> Finding path from Source [S] to Destination [G] using DFS [max depth = 4] ... [S] -> [p] -> [q] -> [e] -> [h] -> [p] Destination [G] NOT found with given max depth value of 4	<u>Output:</u> Finding the Path from Source [S] to Destination [G] using IDS [max depth = 4] ... At Depth 0: [S] At Depth 1: [S] -> [p] -> [e] -> [d] At Depth 2: [S] -> [p] -> [q] -> [e] -> [h] -> [r] -> [d] -> [b] -> [c] At Depth 3: [S] -> [p] -> [q] -> [e] -> [h] -> [p] -> [q]

	<p>-> [r] -> [f] -> [d] -> [b] -> [a] -> [c] -> [a]</p> <p>At Depth 4: [S] -> [p] -> [q] -> [e] -> [h] -> [p] -> [q] -> [q] -> [r] -> [f] -> [c] -> [G]</p> <p>Destination [G] Found at depth 4</p>
1. DFS is performed upto the max_depth value.	1. DFS is performed for each depth \leq max_depth. So all nodes upto max_depth is visited.
2. DFS is performed only once, starting from source node. Thus each node can be visited atmost once.	2. DFS is performed at each node having depth \leq max_depth, each time from the source node. Thus, nodes at shallow levels may be visited multiple times by the algorithm.
3. A node having length(path from source) \leq max_depth may still be unvisited. Further, if length(path) $<$ max_depth, DFS may not yield in shortest path.	3. Every node having length(path from source) \leq max_depth will be visited, and that path will be shortest (since IDS performs level-order DFS). Thus, for length(path) $<$ max_depth, the shortest path will be yielded.