

Week 6:

- I. Given a (directed/undirected) graph, design an algorithm and implement it using a program to find if a path exists between two given vertices or not. (Hint: use DFS)

Input Format:

Input will be the graph in the form of adjacency matrix or adjacency list.

Source vertex number and destination vertex number is also provided as an input.

Output Format:

Output will be '**Yes Path Exists**' if path exists, otherwise print '**No Such Path Exists**'.

Sample I/O Problem I:

Input: 5 0 1 1 0 0 1 0 1 1 1 1 1 0 1 0 0 1 1 0 1 0 1 0 1 0 1 5	Output: Yes Path Exists
--	-----------------------------------

- II. Given a graph, design an algorithm and implement it using a program to find if a graph is bipartite or not. (Hint: use BFS)

Input Format:

Input will be the graph in the form of adjacency matrix or adjacency list.

Output Format:

Output will be '**Yes Bipartite**' if graph is bipartite, otherwise print '**Not Bipartite**'.

Sample I/O Problem II:

Input: 5 0 1 1 0 0 1 0 1 1 1 1 1 0 1 0 0 1 1 0 1 0 1 0 1 0	Output: Not Bipartite
---	---------------------------------

- III. Given a directed graph, design an algorithm and implement it using a program to find whether cycle exists in the graph or not.

Input Format:

Input will be the graph in the form of adjacency matrix or adjacency list.

Output Format:

Output will be '**Yes Cycle Exists**' if cycle exists otherwise print '**No Cycle Exists**'.

Sample I/O Problem III:

Input: 5 0 1 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0	Output: No Cycle Exists
---	-----------------------------------