



ATMA RAM SANATAN DHARM COLLEGE

Course Title:

Discrete Mathematical Structure
Practical

Submitted To:

Shalini Ma'am
Faculty Of Computer Science

Submitted By:

Name : Sudeep Kumar Singh
Roll No. : 22/28021
Course : B.Sc. Computer Science Hons.

7. Write a Program to check if a given graph is a complete graph. Represent the graph using the AdjacencyList representation.

Code:

```
7.py > ...
1  # defining the is_complete_graph function with two parameters graph and num_vertices.
2  def is_complete_graph(graph, num_vertices):
3      # Check if every pair of vertices is connected
4      for i in range(num_vertices):
5          for j in range(num_vertices):
6              if i != j and j not in graph[i]:
7                  return False
8      return True
9
10 def main():
11     # initialising graph dict where keys will be vertices and values will be neighbors
12     graph = {}
13     num_vertices = int(input("Enter the number of vertices in the graph: "))
14
15     # Taking input for the graph in adjacency list form
16     print('\nNote: Here whole numbers are used to represent vertices\n')
17     print("Enter the graph in adjacency list")
18     for i in range(num_vertices):
19         vertex = input(f"Enter the vertices connected to vertex {i}, separated by commas: ")
20         vertices = [int(v) for v in vertex.split(",")]
21         graph[i] = vertices
22
23     print("\nAdjacency list representation:")
24     for vertex, neighbors in graph.items():
25         print(f"{vertex}: {neighbors}")
26
27     if is_complete_graph(graph, num_vertices):
28         print("The graph is a complete graph.")
29     else:
30         print("The graph is not a complete graph.")
31
32 if __name__ == "__main__":
33     main()
```

Output: For complete graph

```
PS C:\Users\Sudeep\OneDrive - RAJDHANI COLLEGE\Desktop\DSA> & C:/Users/Sudeep/DSA/7.py"
Enter the number of vertices in the graph: 4

Note: Here whole numbers are used to represent vertices

Enter the graph in adjacency list
Enter the vertices connected to vertex 0, separated by commas: 1,2,3
Enter the vertices connected to vertex 1, separated by commas: 0,2,3
Enter the vertices connected to vertex 2, separated by commas: 0,1,3
Enter the vertices connected to vertex 3, separated by commas: 0,1,2

Adjacency list representation:
0: [1, 2, 3]
1: [0, 2, 3]
2: [0, 1, 3]
3: [0, 1, 2]
The graph is a complete graph.
```

Output: For not-complete graph

```
PS C:\Users\Sudeep\OneDrive - RAJDHANI COLLEGE\Desktop\DSA> & C:/Users/Sudeep/DSA/7.py"
Enter the number of vertices in the graph: 4

Note: Here whole numbers are used to represent vertices

Enter the graph in adjacency list
Enter the vertices connected to vertex 0, separated by commas: 1,2,3
Enter the vertices connected to vertex 1, separated by commas: 0,2,3
Enter the vertices connected to vertex 2, separated by commas: 0,1,3
Enter the vertices connected to vertex 3, separated by commas: 0,1

Adjacency list representation:
0: [1, 2, 3]
1: [0, 2, 3]
2: [0, 1, 3]
3: [0, 1]
The graph is not a complete graph.
```