Practice Lessons

Nov. 27 2025

1). Adjacency Matrix Representation

- **Input:** The input consists of multiple lines.
  - Each line contains two integers *u* and *v*, indicating that there is an undirected edge between vertices *u* and *v*.
  - Please refer to the sample input for the exact format.
- **Output:**
  - (1) Output the adjacency matrix. There should be a space between every pair of numbers on the same line.
  - (2) Output the adjacency list. There should be a space between a number and "->", and also a space between "->" and "end".
  - There must be a blank line between the printed matrix and the printed list. Refer to the sample output for the exact format.

Sample Input & Output

- Sample input:

```
0 3
1 2
1 3
2 3
```

- Sample output:

```
Adjacency matrix:
0 0 0 1
0 0 1 1
0 1 0 1
1 1 1 0

Adjacency list:
0: 3 -> end
1: 2 -> 3 -> end
2: 1 -> 3 -> end
3: 0 -> 1 -> 2 -> end
```

2). Graph Connectivity

- **Input:** The input consists of two parts.
  - One line containing an **integer** between 1 and 999, representing the number of vertices in the graph.
  - The **adjacency matrix**. There are exactly *n* rows, each containing *n* integers, either 0 or 1, separated by spaces.
    - The *j*-th number in the *i*-th row indicates whether there is an edge from vertex *i* to vertex *j* (0: does not exist, 1: exists). There are no self-loops, so the *i*-th number in the *i*-th row must be 0. The edges are undirected, so the adjacency matrix is symmetric.
- **Output:** If the graph is connected, output `1`; otherwise, output `-1`.

Sample Input & Output

- Sample input:

```
4
0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0
```

- Sample output:

```
1
```

NOTE: In a graph, if for **every** pair of vertices there is a **path** connecting them, the graph is called connected.