C1 - Maximum Flow - The Basic Version

Time Limit: 1 sec.

Problem Description

Implement the Ford-Fulkerson Algorithm and solve the max-flow problem for a given undirected network $G = (V, E)$ with source s and sink t.

Input Format

The first line consists of two integers n and m, the number of vertices in V and the number of edges in E. The second line consists of two integers s and t, the source and sink vertices. Each of the following m lines consists of three integers u, v, c, which indicates that there is an edge between vertex u and vertex v and has capacity w.

You may assume that

- The vertices are numbered from 0 to $n - 1$.
- $2 \leq n \leq 100$.
- $0 \leq m \leq 1000$.
- The capacity of the edges is between 0 and 100.

Output Format

Print the value of the maximum s-t flow.

Sample Input

```
4 3
1 3
0 1 2
2 3 1
1 2 2
```

Sample Output

```
1
```

Note.

This is the basic version of C3, and it is okay if you directly submit the program for C3 to this problem.

C2 - Maximum Flow

Time Limit: 1 sec.

Problem Description

Implement the Ford-Fulkerson Algorithm and solve the max-flow problem for a given undirected network $G = (V, E)$ with source s and sink t.

Input Format

The first line consists of two integers n and m, the number of vertices in V and the number of edges in E. The second line consists of two integers s and t, the source and sink vertices. Each of the following m lines consists of three integers u, v, c, which indicates that there is an edge between vertex u and vertex v and has capacity w.

You may assume that

- The vertices are numbered from 0 to $n - 1$.
- $2 \leq n \leq 100$.
- $0 \leq m \leq 1000$.
- The capacity of the edges is between 0 and 10^7.

Output Format

Print the value of the maximum s-t flow.

Sample Input

```
4 3
1 3
0 1 2
2 3 1
1 2 2
```

Sample Output

```
1
```

Note.

Be careful of **integer overflow** problem. This is the basic version of C3, and it is okay if you directly submit the program for C3 to this problem.

C3 - The Max-Flow Min-Cut Theorem

Time Limit: 1 sec.

Problem Description

Implement the Ford-Fulkerson Algorithm. Compute the max-flow and a min-cut for a given undirected network $G = (V, E)$ with source s and sink t.

Input Format

The first line consists of two integers n and m, the number of vertices in V and the number of edges in E. The second line consists of two integers s and t, the source and sink vertices. Each of the following m lines consists of three integers u, v, c, which indicates that there is an edge between vertex u and vertex v and has capacity w.

You may assume that

- The vertices are numbered from 0 to $n - 1$.
- $2 \leq n \leq 100$.
- $0 \leq m \leq 1000$.
- The capacity of the edges is between 0 and 10^7.

Output Format

Print the value of the maximum s-t flow in the first line. In the following lines, print the endpoints of the edges in a minimum s-t cut, separated by a space, one edge per line.

If there are multiple answers, you can print any of them.

Sample Input

```
4 4
0 3
0 1 2
0 2 1
1 3 2
2 3 2
```

Sample Output

```
3
0 1
0 2
```

Note.

Be careful of **integer overflow** problem. This is the basic version of C3, and it is okay if you directly submit the program for C3 to this problem.

C4 - Max-Flow Min-Cut - Adv Version

Time Limit: 1 sec.

Problem Description

Implement capacity scaling or Edmond-Karp Algorithm. Compute the max-flow and a min-cut for a given undirected network $G = (V, E)$ with source s and sink t.

Input Format

The first line consists of two integers n and m, the number of vertices in V and the number of edges in E. The second line consists of two integers s and t, the source and sink vertices. Each of the following m lines consists of three integers u, v, c, which indicates that there is an edge between vertex u and vertex v and has capacity w.

You may assume that

- The vertices are numbered from 0 to $n - 1$.
- $2 \leq n \leq 500$.
- $0 \leq m \leq 5000$.
- The capacity of the edges is between 0 and 10^{16}.

Output Format

Print the value of the maximum s-t flow in the first line. In the following lines, print the endpoints of the edges in a minimum s-t cut, separated by a space, one edge per line.

If there are multiple answers, you can print any of them.

Sample Input

```
4 4
0 3
0 1 2
0 2 1
1 3 2
2 3 2
```

Sample Output

```
3
0 1
0 2
```