ACM-ICPC Live Archive

6112 Critical 3-Path

The PERT (Project Evaluation and Review Technique) chart, a graphical tool used in the field of project management, is designed to analyze and represent the set of tasks involved in completing a given project. Edges in PERT chart represent tasks to be performed, and edge weights represent the length of time required to perform the task. For vertices u, v, w of a PERT chart, if edge (u, v) enters vertex v and edge (v, w) leaves v, then task (u, v) must be performed prior to task (v, w). A path through a PERT chart represents a sequence of tasks that must be performed in a particular order. Note that there is no cycle in the PERT chart. A critical path is a longest path in PERT chart, corresponding to the longest time to perform an ordered sequence of tasks. The weight of a critical path is a lower bound on the total time to perform all the tasks in a project.

A *3-path* of six distinct vertices $s_1, s_2, s_3, t_1, t_2, t_3$ in a PERT chart is defined as follows:

(1) A 3-path consists of three paths P_i from vertex si to vertex ti for $i = 1, 2, 3$.

(2) The paths P_1, P_2, P_3 are vertex-disjoint, i.e., no two of the paths have vertices in common.

The length of a 3-path is the sum of the length of the 3 paths P_1, P_2, P_3. A *critical 3-path* of six distinct vertices in a PERT chart is a 3-path of maximum length over all 3-paths.

For example, a critical 3-path $\{P_1, P_2, P_3\}$ of a graph in Figure 1, where P_1 is a path from vertex 3 to vertex 15, P_2 is a path from vertex 4 to vertex 16, and P_3 is path from vertex 5 to vertex 17, is as follows:

$$P_1 \ : \ 3 \to 6 \to 11 \to 15$$
$$P_2 \ : \ 4 \to 7 \to 9 \to 12 \to 16$$
$$P_3 \ : \ 5 \to 8 \to 13 \to 17$$

The length of the critical 3-path is 128.

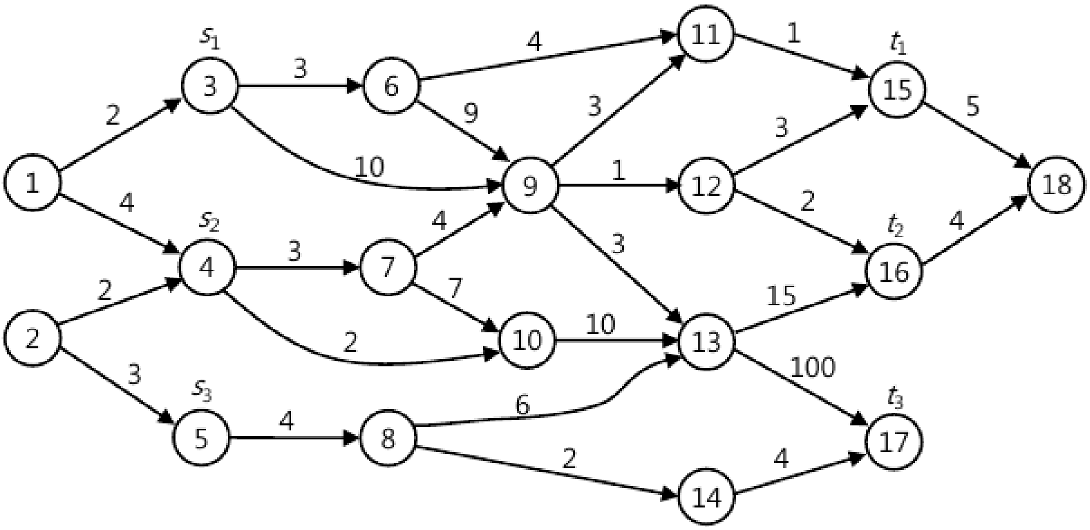

Figure 1. A sample PERT chart

Given a graph corresponding to a PERT chart and six distinct vertices, write a program to find the length of critical 3-path of the graph corresponding to the six vertices.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing two integers, n and m ($6 \le n \le 100, n-1 \le m \le n(n-1)/2$), where n is the number of vertices and m is the number of edges. Every input node is numbered from 1 to n. Next line contains six integers $s_1, s_2, s_3, t_1, t_2, t_3$, where all six integers are distinct. In the following m lines, the weight of the directed edges are given; each line contains three integers, u, v, and W ($1 \le W \le 100,000$), where W is the weight of an edge from vertex u to v. You may assume that $u < v$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the length of critical 3-path P_1, P_2, P_3, where P_i is a path from s_i to t_i ($1 \le i \le 3$). If there does not exist a critical 3-path, print '0'.

The following shows sample input and output for two test cases.

Sample Input

```
2
18 27
3 4 5 15 16 17
1 3 2
1 4 4
2 4 2
2 5 3
3 6 3
3 9 10
4 7 3
4 10 2
5 8 4
6 11 4
6 9 9
7 9 4
7 10 7
8 13 6
8 14 2
9 11 3
9 12 1
9 13 3
10 13 10
11 15 1
12 15 3
12 16 2
13 16 15
13 17 100
14 17 4
15 18 5
16 18 4
6 5
1 2 3 4 5 6
1 2 1
```

```
2 3 1
3 4 1
4 5 1
5 6 1
```

Sample Output

```
128
0
```