**Problem D. Flood**

In one town there are n junctions, some of which are connected by direct two-way streets. Between two junctions there can be more than one direct street. In spring there are huge floods because of the oncoming river and some streets are flooded. For each street the time, needed for the municipal company to clean it, is known. Some of the flooded streets have to be cleaned from the river’s water, so that it is possible to reach each junction from any other. The task of the municipal company is to ensure that each junction can be reached from any other for a minimum amount of time.

Write a program, called **flood**, which finds the minimum time for which the company will cope with the task by given *n* junctions and *m* streets. The program should work for several test cases.

**Input**

The first line of the standard input contains the number of test cases. For each test case there are several more lines. The first consists of three numbers *n*, *m* and *k,* respectively the number of junctions, the total number of streets and the number of flooded streets. Every line of the next *m* lines contains three numbers (*xi, yi, ti)* – the junctions connected with a direct street and the time for which the company will clean it, if it’s flooded. The next line contains the streets which are flooded. The streets are numbered from 1 to *m* in the order they were entered.

**Output**

For each test case in one line of the standard output the program should print the minimum time for which the company will cope with the task.

**Limitations**

1 ≤ *n* ≤ 1000

1 ≤ *m* ≤ 10000

1 ≤ *k* ≤ *m*

1 ≤ *ti* ≤ 30000

**Example**

**Input**

1

7 8 4

0 2 5

2 1 2

1 3 8

3 4 12

6 4 1

5 4 10

6 5 4

2 3 6

1 4 6 7

**Output**

21