Tree-SAT

Output Only Task

Statement

*Input files are available in **zip format** for Unix (Linux/macOS) and Windows. Go to the attachments section on the judge/contest system to download them.*

You have a weighted undirected graph with N nodes and M edges, with nodes numbered 1 through N and edges numbered 1 through M. There can be multiple edges connecting the same pair of nodes, but there will never be an edge that connects a node to itself. Find a subset of edges in the graph such that the graph is connected through only those edges, while keeping the sum of the weights of the edges in the set as small as possible.

Wait... It's not that simple. You are also given K constraints. Each constraint is of one of the below types:

Type 1: Given two edges a, b, either both the edges must be used, or neither can be used.

Type 2: Given two edges a, b, at most 1 of the edges can be used.

Subject to these constraints, find a suitable set of edges that connect the graph, while keeping the sum of edge weights small. You are guaranteed that such a set exists. Note that your output **does not necessarily have to form a tree**.

Input

You are given 10 input files (`input_*.txt`), each worth 10 points. Each file describes an input scenario in the following manner:

- The first line contains 3 integers N M K.
- The next M lines each contain 3 integers, u_i v_i w_i, denoting that the ith edge connects nodes u_i and v_i and has cost w_i.
- The next K lines each contains 3 integers, each representing a constraint: t_i a_i b_i where t_i denotes the type of the constraint and a_i, b_i are as above.

Output

- Provide up to 10 output files (`output_*.txt`) corresponding to the input files.
- The first line should contain a single integer E representing the total number of edges you use.
- The next line should contain E space-separated integers, representing the numbers of the edges you use in your solution, **in ascending order**.

Bounds

All input files adhere to the following bounds.

- $1 \leq N \leq 1000$
- $1 \leq M \leq 10^5$
- $0 \leq K \leq 10^5$

Scoring

For each test case there are 2 values X and P. Let S be the total sum of edge weights of your solution. Then the score for a solution is given by:

$$\begin{cases} 0 & \textit{Solution is invalid} \\ \min\left(10, 3 + 7 \times e^{\frac{1.25 \times \left(\frac{S}{X} - 1\right)}{1 - P}}\right) & S > X \\ 10 & S \leq X \end{cases}$$

In other words, valid solutions are scored on an exponential scale. Notably:

- If $S \leq X$, then 10 points will be awarded for a valid solution.
- If $S = P \times X$, then 5 points will be awarded
- You will always be awarded at least 3 points for any valid solution.

The values of X and P are, for each test case:

Input file	X	P
`input_01.txt`	2865985	1.2
`input_02.txt`	1246408	1.2
`input_03.txt`	249962463	1.01
`input_04.txt`	245819044	1.01
`input_05.txt`	92744414	1.1
`input_06.txt`	24885852	1.5
`input_07.txt`	77498535	1.1
`input_08.txt`	56764260	2.0
`input_09.txt`	47866077	1.5
`input_10.txt`	8597955	1.001

Note that a solution will score **zero points** if any of the below conditions are true:

- The selected edges do not ensure the graph is connected, or one of the given constraints are violated.
- The edges are not output in ascending order.
- An edge is output twice.
- You violate any of the K constraints.

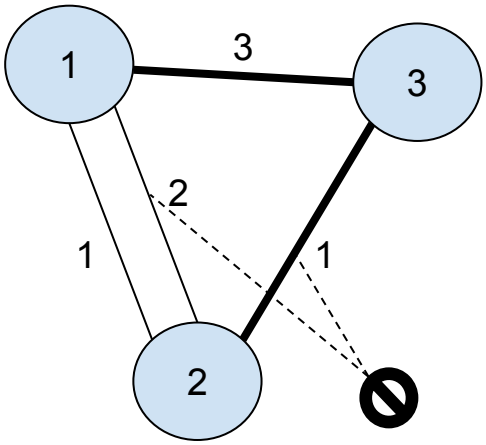
Sample Input

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

Sample Output

```
2
3 4
```

Explanation



The above diagram illustrates the sample input. The two bold edges must be taken together, and the two edges that cannot be taken together are marked.

- The optimal set of edges to select to make sure the graph is connected are the third and the fourth, that is, the edge from node 1 to node 3 and the edge from node 2 to node 3.
- This is because you cannot take the second edge and the fourth edge together, and when you take the fourth edge, you have to take the third edge.
- Note that your solutions do not necessarily have to be optimal.

Submitting

To submit your output files, you must do one of the following:

Submitting multiple test cases

- Create a zip file containing all of your output files. You may include as many or as few output files as you like (i.e., you do not need a solution for every input scenario). On a GNU/Linux system you can use a command like the following:

  ```
  zip mysolutions.zip output_*.txt
  ```

  On Windows systems you can create a zip file by selecting *File* → *New* → *Compressed (zipped) Folder* from within Windows Explorer, and then you can copy your output files into this new zip file.

- Submit this zip file to the judge or contest system, in the same way that you would submit an ordinary solution.

Submitting individual test cases

- If you are attempting this problem on *Contest Management System*, such as during the live contest, you will be able to submit individual output files through the contest system itself. There should be upload buttons for each individual output file.

- Once submitting an output file, the contest system will merge the output file with any previous output files you have submitted, and judge them altogether. If you are submitting an output file that has already been submitted, the older file will be overwritten.