ETH Zurich
Institut für Theoretische Informatik
Prof. Dr. Angelika Steger

ACM Programming Challenges Lab

Exercise 1 – *Island Telecom*

Description You are a young manager working for a telecommunications company serving many small and large island-states in the pacific. Your boss tells you that the company wishes to expand to new islands, and that you should figure out the cost of building an optimal network on each island. He hands you a data-set containing a long list with cost estimates. Each item on the list consists of two villages and an estimate of how much the company would need to spend to put a direct network link between them. You also get these instructions:

- You can think of this list as a list of weighted edges of a graph where each village is a vertex.
- The data-set contains information about many islands but you don't know which villages are on which island.
- However you know that there are enough edges such that the villages of each island form a connected component in this graph. There are also *no* edges between villages on different islands.
- The network your company wishes to build on each island corresponds to the cheapest connected subgraph that covers *all* villages on that island.
- The cost of a network is simply the sum over the costs of all edges that are in your subgraph. Islands with only one village are not interesting to your company and you should ignore them.
- The company does not want to spend too much for the network on each island. The budget for each island is capped at 1'000'000.

You dust off your old algorithm book from university and start working on a program to compute what your boss wants.

Input The first line of input contains the number of data-sets to parse. The first line of each data-set contains two positive integers separated by a space. The first one is $1 \leq n \leq 10'000$, the number of villages in the data-set. The second one is $m \geq 0$ and gives you the number of cost estimates in the data set. Then m lines follow. On each line i there are three integers u_i, v_i and c_i. The integers u_i and v_i are an ID of two villages and $c_i > 0$ denotes the cost estimate for connecting u_i and v_i. The IDs are all in the range 0 to $n - 1$ and no estimate appears twice, not even in its reverse form (with u_i and v_i flipped). It also holds that the sum of all c_i (within one data-set) is not larger than 2^{30}. IDs which never appear in these m lines correspond to islands with just one village on them.

Output For each data-set you need to compute the cost of the cheapest network on all islands. For each data-set output a single line containing two integers separated by a space. The first integer is the number of islands on which the cheapest possible network costs at most 1'000'000. The second is the cost of the cheapest communication network over all islands. Note that the first integer is 0 if and only if the second is larger than 1'000'000.

Test Sets

- **Simple (40 points)** For this part you can assume that there is exactly 1 island in each data-set (in other words, the edges in the input form a connected graph). Note that the last two cases in the sample input below do not satisfy this restriction.
- **Hard (60 points)** This test-set is large and the number of islands is not restricted.

Implementation note The test-sets for this exercise can be quite large and the overhead of just reading the test set can be large as well. If you use `std::cin` to read the input put the following line at the beginning of your `main()` function for a significant performance boost:

```
std::ios_base::sync_with_stdio(false);.
```

If you do this however, you must not use any of the IO functions defined in the header file `cstdio`, such as `printf` or `scanf`.

If you want to use `printf` and `scanf` (or other `cstdio` functions) you *must not* include the mentioned line in your `main()` function!

Sample input

```
3
3 3
0 1 100
0 2 400
2 1 150
4 2
0 1 1000001
2 3 647
4 2
0 2 5432
1 3 1234
```

Sample output

```
1 250
1 647
2 1234
```