Eidgenössische
Technische Hochschule
Zürich

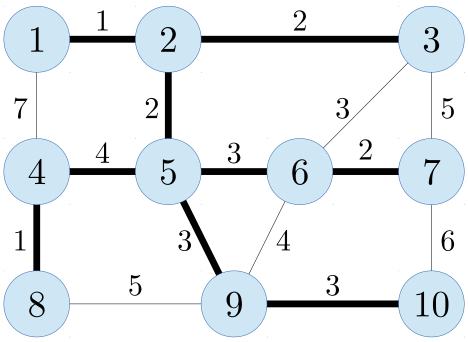
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Institut für Theoretische Informatik
Peter Widmayer
Sandro Montanari
Tobias Pröger

24th April 2013

Datenstrukturen & Algorithmen   Programming Exercise 9   FS 13

In this exercise we are going to implement Kruskal's algorithm for the *minimum spanning tree problem*. This problem is defined as follows. Given an undirected graph $G = (V, E)$ with the edge cost function $c : E \to \mathbb{Q}^+$, we search for an acyclic subset $T \subseteq E$ that connects all vertices in V (i.e., a tree) and whose total cost $\sum_{e \in T} c(e)$ is minimum among all possible trees. The following image shows an example where the bold edges form a minimum spanning tree of the graph.



Input The first line contains only the number t of test instances. After that, we have exactly one line per test instance containing the description of the input graph $G = (V, E)$ in the form $n, m, u_1, v_1, c_1, ..., u_m, v_m, c_m$. We have $1 \leq n, m \leq 10000$ with $V = \{1, ..., n\}$ and $|E| = m$. For every i, $1 \leq i \leq m$, the numbers $u_i, v_i \in \{1, ..., n\}$ define the edge $\{u_i, v_i\} \in E$ having a cost of c_i, $1 \leq c_i \leq 1000$.

Output For every test instance we output only one line. This line contains the cost of a minimum spanning tree.

Example

Input:

```
1
10 15 1 2 1 2 3 2 1 4 7 2 5 2 3 6 3 3 7 5 4 5 4 5 6 3 6 7 2 4 8 1 5 9 3 6 9 4 7 10 6 8 9 5 9 10 3
```

Output:

```
21
```

Hand-in: until Wednesday, 8th May 2013.