Algorithm Design Laboratory with Applications

Prof. Stefano Leucci

Problem: *Gift.*

Alice and Bob are invited to the birthday party of Charlie. Alice, Bob, and Charlie all live in the same city, which is modeled as a connected undirected edge-weighted graph G, whose vertices represent locations and are indexed with the integers from 0 to $n-1$, and whose m edges represents roads and are weighted with the respective road length (weights are non-negative integers). Alice's home is at vertex v_A, Bob's home is at vertex v_B, and the location of the party is vertex v_C. Alice and Bob independently want to buy a present for Charlie, and then they need to reach v_C. The gift shops are located in k distinct vertices $s_1, s_2, \ldots, s_k$ of G and buying a present from s_i costs c_i dollars. In order to save money on gas, Alice and Bob agree that, after having bought their gifts, they will meet in some vertex v_M and then travel together from v_M to v_C in one of their cars. In this way Alice and Bob independently pay for the gas needed to reach v_M and for the cost of their respective gifts, and then they share the cost of the gas needed to travel from v_M to v_C. Alice's and Bob's car both consume g dollars of gas per unit of length.

Your task is to write an algorithm that finds the best meeting vertex v_M in order to minimize the *total amount* τ paid by Alice and Bob (this includes both the overall gas cost and amount paid to buy the two gifts).

Input. The input consists of a set of instances, or *test-cases*, of the previous problem. The first line contains the number T of test-cases. The first line of each test-case contains the seven integers n, m, k, v_A, v_B, v_C, and g. The i-th of the following k lines describes the i-th gift shop and contains the integers v_i and c_i. The final m lines each describe one of the edges of G: each line contains three integers u, v, w, to signify that G contains the undirected edge (u, v) with weight w.

Output. The output consists of T lines. The i-th line is the answer to the i-th test-case and contains two integers v_M and τ. Here v_M is the best meeting point from Alice and Bob (i.e., the one that minimizes the final cost τ) and τ is the overall amount paid by Alice and Bob if they meet in v_M. In cases of ties, prefer the vertex v_M with the smallest index.

Assumptions. $1 \leq T \leq 10$; $1 \leq n \leq 2^{16}$; $1 \leq m \leq 2^{16}$; $1 \leq g \leq 2^8$;
$1 \leq k \leq n-3$; $\forall i = 1, \ldots, k, \leq s_i \in \{0, \ldots, n-1\} \setminus \{v_A, v_B, v_C\}$; $\forall i = 1, \ldots, k$, $1 \leq c_i \leq 2^{12}$;
The edge weights are integers in $\{1, \ldots, 2^{10}\}$.

(Continues on the next page)

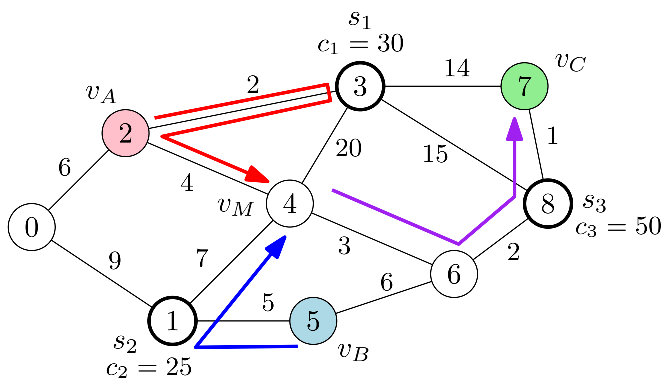

Figure 1: An example instance for $g = 2$. v_A is in red, v_B is in blue, v_C is in green, and shops are shown in bold. In an optimal solution Alice and Bob meet in vertex $v_M = 4$. Alice buys her gift in shop s_1 on vertex 3, and Bob buys his gift in shop s_2 on vertex 1. The cost for Alice to buy the gift and reach v_M is $2(2 + 2 + 4) + 30 = 46$. The corresponding cost for Bob is $2(5 + 7) + 25 = 49$. The cost shared between Alice and Bob is $2(3 + 2 + 1) = 12$. The total cost is $\tau = 46 + 49 + 12 = 107$.

Example.

Input (corresponding to Figure 1):

```
1
9 13 3 2 5 7 2
3 30
1 25
8 50
0 1 9
0 6 2
1 4 7
1 5 5
2 3 2
2 4 4
3 4 20
3 7 14
3 8 15
4 6 3
5 6 6
6 8 2
7 8 1
```

Output:

```
4 107
```

Requirements. Your algorithm should require time $O(m + n \log n)$ (with reasonable hidden constants).

Notes. A reasonable implementation should not require more than 1 second for each input file. The vertices v_A, v_B, v_C, and v_M do not necessarily need to be distinct.