Museum

Time limit: 1 second
Memory limit: 1024 MB

Elena works as a guide at the Kaunas City Museum. The museum consists of N rooms numbered from 1 to N, and M two-way corridors between them. It is guaranteed that there are no corridors connecting a room to itself, and every pair of rooms is connected by at most one corridor. The time required to traverse each corridor is known.

Each day, Elena gives at most K tours. Each tour starts from the entrance (room 1) and finishes at the exit (room N). The total time of the tour is equal to the sum of the times for passing the corridors.

Elena feels bored showing things in the same order, so this is how she picks her tours:

- She chooses a shortest tour for the first group.
- For the next group, she chooses the next shortest tour different from the previous one, and so on.

Two tours differ if the rooms visited differ in the tour, or the sequence in which the rooms were visited differ. Note that a tour may revisit rooms, so there are infinitely many possible tours between the entrance in the room 1 and exit in the room N.

The museum recently went through a refurbishment and Elena lost her tours. Now Elena asks for your help to find the K shortest tours from room 1 to room N. The tours must be different.

Implementation details

You should implement the function `find_tours`:

```
std::vector<int> find_tours(int N, int M, int K,
                            std::vector<std::pair<int, int>> e,
                            std::vector<int> t)
```

- N: the number of rooms in the museum;
- M: the number of corridors between the rooms;
- K: the number of tours;
- e: a vector of M pairs, representing the corridors;

- t: a vector of M positive integers, representing the times for the corresponding corridors.

This function will be called once for each test and has to return a vector with K numbers – the total times for the K shortest tours from room 1 to room N in non-decreasing order.

Constraints

- $2 \leq N \leq 10\,000,\ 1 \leq M \leq 20\,000,\ 1 \leq K \leq 75$
- $1 \leq t[i] \leq 200\,000$ for each $1 \leq i \leq M$.

Example

Consider the following call for $N = 3$, $M = 3$, and $K = 3$:

```
find_tours(3, 3, 3, {{1, 2}, {2, 3}, {1, 3}}, {3, 2, 5})
```

The call should return `{5, 5, 9}`, since the 3 shortest tours between room 1 and room 3 are:

- $1 - 2 - 3$ (with corresponding corridor times 3 and 2) for a total time of 5;
- $1 - 3$ (with corresponding corridor time 5) for a total time of 5;
- $1 - 2 - 3 - 2 - 3$ (with corresponding corridor times 3, 2, 2, 2) for a total time of 9.

Subtasks

Subtask	Points	N	M	K	Additional constraints
0	0	–	–	–	The example.
1	13	≤ 10	≤ 45	≤ 75	None of the shortest K tours revisit rooms.
2	28	$\leq 10\,000$	$\leq 20\,000$	$= 1$	–
3	59	$\leq 10\,000$	$\leq 20\,000$	≤ 75	–

Sample grader

The input format is the following:

- line 1: three integers – the values of N, M, and K.
- line $1 + i$: three integers x_i, y_i, t_i – representing a corridor between the rooms x_i and y_i with passing time equal to t_i.

The output format is the following:

- line 1: the total times for the tours as returned by the call.