(Adv.) Competitive Programming

Submit until end of contest, via the judge interface

Problem: speedrun (1 second timelimit)

Recently, dedicated speedrunners found a new exploit in your favorite game of all time *The Legend of Ganondorf: Ocarina Dreams*. The game is about playing the organ for princess Zelda, while wondering if you should have picked up this cool ocarina she dropped on her way to your castle.

The glitch speeds up several sections of the game and even introduces some new sequence breaks. With all these changes to the established timings and possible routes, runners need to figure out the optimal way to the credits again. You decide to put your programming skills to good use. The game can be modeled as a directed multi graph, where nodes are milestones such as events, cutscenes, items locations, dungeons, or boss fights. Edges have a weight, indicating the time from one milestone to another.

500k sparse -> adjacency list 10000

Input The first line contains the number of nodes $0 < n \leq 10^4$, and the number of edges $m \leq 50n$. The next m lines each contain three integers: the endpoints of a directed edge $0 \leq u, v < n$ and a weight $1 \leq w \leq 10^6$. The game always starts with Ganondorf chasing Zelda and ends with the credits. These two events have ids 0 and $n-1$, respectively. It goes without saying that the game is possible.

Output In the first line, print the duration of the fastest route through the game. In the second line, print the milestones of the route separated by spaces. If there are multiple routes with the same time, print the lexicographically smallest one to keep it close to the way the game was designed.

Sample input

```
5 6
0 1 2
0 2 2
0 3 3
1 4 3
2 4 2
3 4 2
```

Sample output

```
4
0 2 4
```

```
4 4
0 1 2
0 2 1
1 3 1
2 3 2
```

```
3
0 1 3
```

```
5 5
0 2 2
0 3 1
3 1 1
2 4 1
1 4 1
```

```
3
0 2 4
```