

Задача 6. Фиолетовое такси

Источник:	основная II
Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

«Фиолетовое такси» предложило клиентам новую услугу — теперь можно узнать кратчайшее время проезда для любых мест отправления и назначения. Однако диспетчеры не успевают отвечать на все запросы. Напишите программу, которая поможет диспетчерам быстро находить минимальное время пути. Учтите, что в редких случаях клиента интересует не только время, но и подробное описание предполагаемого маршрута.

Формат входных данных

В первой строке входного файла записано четыре целых числа: N — количество пунктов, в которых люди садятся и выходят из такси, M — количество дорог, их соединяющих, P — количество запросов на поиск кратчайшего пути, K — количество запросов на поиск минимального времени ($1 \leq N, P \leq 300$, $1 \leq M, K \leq 50\,000$).

Далее в M строках описываются дороги, по три целых числа в каждой строке — номера двух пунктов, соединенных этой дорогой и время L_i проезда по дороге ($1 \leq L_i \leq 10^6$).

Пункты пронумерованы числами от 1 до N . Все дороги двусторонние. Пару пунктов **может** соединять несколько дорог. Гарантируется, что все пункты соединены между собой.

Далее следуют запросы, по одному запросу в строке. Сначала записаны запросы на поиск кратчайшего пути (P штук), затем запросы на поиск минимального времени (K штук). Каждый запрос описан двумя целыми числами S_j и T_j — номер пункта отправления и номер пункта назначения соответственно ($1 \leq S_j \neq T_j \leq N$).

Формат выходных данных

Для каждого запроса нужно вывести ответ на отдельной строке. Во-первых, нужно вывести минимальное время пути от пункта отправления до пункта назначения. Во-вторых, для запроса на поиск кратчайшего пути нужно вывести дополнительно $(A_j + 1)$ целых чисел — описание оптимального маршрута. Первое из этих чисел должно быть равно A_j — количеству вершин в пути (включая пункты отправления и назначения), а остальные числа должны задавать номера пунктов, по которым идёт маршрут, в порядке их прохождения.

Пример

input.txt	output.txt
5 6 3 2	16 4 3 2 1 5
4 2 2	9 3 1 2 3
1 4 8	9 3 3 2 1
2 3 6	8
1 5 7	12
2 1 3	
4 3 9	
3 5	
1 3	
3 1	
4 3	
5 4	