

Время на тест: 2.0 сек

Ввод: input.txt

Вывод: output.txt

Ограничение по памяти: 131 072 Кб

Куфический дирхем

Петр Васильевич Колошин всегда был коммуникабельным человеком и, соответственно, имел большое количество друзей. Как-то вечером, просматривая свою электронную почту, Петр Васильевич был приятно удивлен приглашением на юбилей своего студенческого друга Андрея Федоровича и сразу же принял положительное решение. Выяснив, что Андрей Федорович уже много лет увлекается коллекционированием старинных монет и охотится за средневековой серебряной монетой с названием "Куфический дирхем", Петр Васильевич решил непременно подарить эту монету.

Воспользовавшись Интернетом, Петр Васильевич сумел узнать все **K** городов, где можно приобрести данную монету, а также ее стоимость в каждом из этих городов. Страна, в которой живут Петр Васильевич и Андрей Федорович, насчитывает **N** городов и **M** двусторонних автомобильных дорог, каждая из которых связывает два различных города государства. Известно, что Петр Васильевич живет в городе **A**, а Андрей Федорович - в городе **B**. Для каждой дороги Петр Васильевич вычислил стоимость проезда с учетом технических характеристик своего автомобиля. В целях экономии Петр Васильевич решил купить монету по пути из города **A** в город **B**. Другими словами, маршрут движения Петра Васильевича должен проходить через город, в котором он решит купить монету. Однако оказалось, что ехать через город, в котором монета стоит меньше всего, не всегда выгодно, так как выиграв в стоимости монеты, можно потерять гораздо больше в стоимости дороги и наоборот: Ваша задача - помочь Петру Васильевичу выбрать оптимальный маршрут и город **Z**, где следует приобрести монету. Маршрут должен начинаться в городе **A**, заканчиваться в городе **B** и проходить через город **Z**. Стоимость данного маршрута должна быть минимальной. Под стоимостью маршрута будем понимать сумму количества денег, потраченных на дорогу и стоимость монеты в городе **Z**. Ниже приведен пример для **N** = 5, **M** = 7, **A** = 1, **B** = 4.

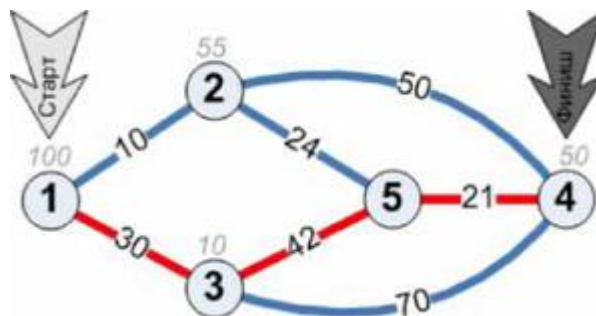


Рисунок 1. Визуализация второго примера.

Для данного примера **K** = 4, стоимость монеты обозначена сверху над кругом, обозначающим город. Оптимальный маршрут выделен красным цветом. Для данного примера **Z** = 3.

Входные данные:

Первая строка входного файла содержит три целых числа **N**, **M** и **K** ($2 \leq N \leq 5000$; $1 \leq M \leq 100000$; $1 \leq K \leq N$), где **N** - количество городов в стране, **M** - количество дорог, а **K** - количество городов, в которых продается искомая монета. Будем считать, что все города пронумерованы целыми числами от 1 до **N**.

Вторая строка входного файла содержит два целых числа **A** и **B** ($1 \leq A, B \leq N$; $A \leq B$), где **A** - номер города, в котором живет Петр Васильевич, а **B** - номер города, в котором живет Андрей Федорович.

Третья строка содержит **K** пар целых чисел **V_i** и **C_i** ($1 \leq V_i \leq N$; $1 \leq C_i \leq 10^9$), где **V_i** - это номер города, в котором можно приобрести искомую монету, а **C_i** - стоимость монеты в

соответствующем городе. Известно, что $V_i \neq V_j$, если $i \neq j$. Все числа в строке разделены одиночными пробелами.

Каждая последующая из M строк содержит три числа X_i, Y_i, S_i ($1 \leq X_i, Y_i \leq N$; $X_i \neq Y_i$; $1 \leq S_i \leq 10^5$), где X_i и Y_i - номера городов, связанных двусторонней дорогой, а S_i - стоимость проезда по данной дороге. Не существует двух различных дорог, связывающих одни и те же города.

Выходные данные:

Единственная строка выходного файла должна содержать одно целое число - минимальную стоимость маршрута. Гарантируется, что решение существует.

input.txt	output.txt	Маршрут, город Z
3 3 2 3 1 1 20 2 5 1 2 7 1 3 5 2 3 8	20	{3, 2, 1} Z = 2
5 7 4 1 4 1 100 4 50 3 10 2 55 1 2 10 5 3 42 1 3 30 2 4 50 3 4 70 2 5 24 4 5 21	103	{1, 3, 5, 4} Z = 3
8 7 1 1 6 5 187 1 8 32 8 6 39 5 4 51 1 4 101 2 4 17 3 7 46 2 8 23	440	{1, 8, 2, 4, 5, 4, 2, 8, 6} Z = 5