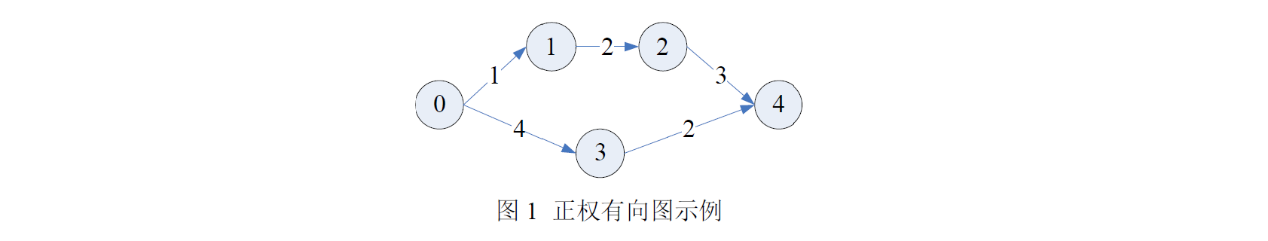
1、给定一个正权有向图，图中包含n个顶点，编号为0至n-1。以顶点0作为源点，编程序求顶点0到各顶点的最短路径。若顶点0到某顶点存在多条最短路径，则输出经过顶点最少的那条路径，例如图1，0到4的最短路径有两条，0→1→2→4和0→3→4，权值和均为6，但0→3→4经过顶点个数最少，故为所求。输入数据保证对每个顶点最多只有一条满足上述条件的路径。



输入格式:

输入第一行为两个正整数n和e，分别表示图的顶点数和边数，其中n不超过15000，e不超过1000。接下来e行表示每条边的信息，每行为3个非负整数a、b、c，其中a和b表示该边的端点编号，c表示权值。各边并非按端点编号顺序排列。

输出格式:

输出为若干行由“->”间隔的数字序列，每行为源点0到某顶点的满足条件的最短路径，如源点到某顶点无最短路径，则不输出该条路径。各条路径按终点的递增顺序输出，源点到源点的最短路径无需输出。

输入样例:

5 5

0 1 1

0 3 4

1 2 2

2 4 3

3 4 2

输出样例:

0->1

0->1->2

0->3

0->3->4

5 8

0 1 2

0 2 6

0 3 9

1 2 3

1 4 7

2 3 1

2 4 2

3 4 2

2、假设有一个公司在某个地区有n个产品销售点，现根据业务需要打算在其中某个销售点上建立一个中心仓库，负责向其它销售点提供产品。由于运输线路不同，运输费用也不同。假定每天需要向每个销售点运输一次产品，那么应将中心仓库建在哪个销售点上，才能使运输费用达到最低。

【问题分析】这是一个求图的中心顶点的问题：即在一个带权图G中，求出这样一个顶点v，使得v到其余顶点的最短路径长度之和最小。首先用弗罗伊德算法求出各个顶点之间的最短路径长度，然后再求出每个顶点到其余各顶点的最短路径长度之和，从中选取一个最短路径长度之和最小的顶点。

输入格式:

输入第一行为两个正整数n和e，分别表示图的顶点数和边数，其中n不超过15000，e不超过1000。接下来e行表示每条边的信息，每行为3个非负整数a、b、c，其中a和b表示该边的端点编号，c表示权值。各边并非按端点编号顺序排列。

输出格式:

输出两个正整数，分别表示中心顶点编号、最小总路径。

输入样例：

6 9

0 1 6

0 2 2

0 3 7

1 2 3

1 4 1

2 3 4

2 5 2

3 5 10

4 5 4

输出样例：

2 15