# 过年回家

## 题目描述

NowCoder今年买了一辆新车，他决定自己开车回家过年。回家过程中要经过n个大小收费站，每个收费站的费用不同，你能帮他计算一下最少需要给多少过路费吗？

### 输入描述:

输入包含多组数据，每组数据第一行包含两个正整数m（1≤m≤500）和n（1≤n≤30），其中m表示接下来输入的行数，n表示有n个收费站，编号依次为1、2、…、n。出发地的编号为0，终点的编号为n，即需要从0到n。

紧接着m行，每行包含三个整数f、t、c，（0≤f, t≤n; 1≤c≤10），分别表示从编号为f的地方开到t，需要交c元的过路费。

### 输出描述:

对应每组数据，请输出至少需要交多少过路费。

### 输入例子:

8 4

0 1 10

0 2 5

1 2 2

1 3 1

2 1 3

2 3 9

2 4 2

3 4 4

### 输出例子:

7

## 解题思路

根据题意，可以根据输入构造一个有向图，其中出发地和收费站表示图的顶点，过路费表示有向边的权重。要求出发地到终点的最少收费，等价于求起点和终点向短路径，可以使用Dijkstra算法进行处理。

### Dijkstra算法

#### 1.定义概览

Dijkstra(迪杰斯特拉)算法是典型的单源最短路径算法，用于计算一个节点到其他所有节点的最短路径。主要特点是以起始点为中心向外层层扩展，直到扩展到终点为止。Dijkstra算法是很有代表性的最短路径算法，在很多专业课程中都作为基本内容有详细的介绍，如数据结构，图论，运筹学等等。**注意该算法要求图中不存在负权边。**

问题描述：在无（有）向图G=(V,E)中，假设每条边E[i]的长度为w[i]，找到由顶点到其余各点的最短路径。（单源最短路径）

#### 2.算法描述

1. **算法思想：**

设G=(V,E)是一个带权有向图，把图中顶点集合V分成两组，第一组为已求出最短路径的顶点集合（用S表示，初始时S中只有一个源点，以后每求得一条最短路径，就将加入到集合S中，直到全部顶点都加入到S中，算法就结束了），第二组为其余未确定最短路径的顶点集合（用U表示），按最短路径长度的递增次序依次把第二组的顶点加入S中。在加入的过程中，总保持从源点v到S中各顶点的最短路径长度不大于从源点v到U中任何顶点的最短路径长度。此外，每个顶点对应一个距离，S中的顶点的距离就是从v到此顶点的最短路径长度，U中的顶点的距离，是从v到此顶点只包括S中的顶点为中间顶点的当前最短路径长度。

1. **算法步骤：**
2. 初始时，S只包含源点，即S＝{v}，v的距离为0。U包含除v外的其他顶点，即:U={其余顶点}，若v与U中顶点u有边，则<u,v>正常有权值，若u不是v的出边邻接点，则<u,v>权值为∞。
3. 从U中选取一个距离v最小的顶点k，把k，加入S中（该选定的距离就是v到k的最短路径长度）。
4. 以k为新考虑的中间点，修改U中各顶点的距离；若从源点v到顶点u的距离（经过顶点k）比原来距离（不经过顶点k）短，则修改顶点u的距离值，修改后的距离值的顶点k的距离加上边上的权。
5. 重复步骤b和c直到所有顶点都包含在S中。

#### 3.举例说明

根据例子的输入构造有向图，然后使用Dijkstra算法求最短路径，路径的构造如图1所示。



图1　Dijkstra构造最短路径