## 关键路径计算

（http://fz.openjudge.cn/graph/015/）

**描述**

给出一个AOE图，请输出其关键路径；

**输入**

第一行：n，m（n,m<=100分别表示节点个数与边的条数）

接下来m行，第i+1行，表示第i条边，每行三个数：ai,bi,ci，表示存在一条ai指向bi权值为ci的边。

注意：起点为结点1，终点为结点n

**输出**

将关键路径的边的编号从小到大输出

**样例输入**

9 11

1 2 6

1 3 4

1 4 5

2 5 1

3 5 1

4 6 2

5 7 9

5 8 7

6 8 4

7 9 2

8 9 4

**样例输出**

1

4

7

8

10

11

# HDU4109 Instrction Arrangement

http://acm.hdu.edu.cn/showproblem.php?pid=4109

**Problem Description**

阿狸这个学期学了计算机组织结构课程。他知道了指令之间可能有依存关系，像WAR（write after read，读之后写），WAW，RAW。如果两个指令之间的时间小于安全时间，它就会导致危险，从而引起错误的结果。所以我们需要设计特殊的回路去消除危险。然而，解决这个问题最简单的方法是添加等待时间（或者由其它的操作填充），如果没有其它的操作填充就意味着需要浪费时间去保证两个指令之间的时间不小于安全时间。

两个指令之间的时间的定义就是它们起始时间差。

现在我们有很多指令，并且我们知道依存关系和指令之间的安全事件。我们也有非常强大的无限核的CPU。所以你可以在同一时间想跑多少指令就跑多少指令。这个CPU可以只消耗1ns就完成任何指令。

你的工作就是重新安排这些指令顺序所以CPU就可以用最短时间完成所有的指令。

**Input**

输入包括几个测试案例

头两行有两个整数N,M（N<=1000,M<=1000），其中N表示N条指令和M表示M个依存关系

接下来的M行，每行包括3个整数X,Y,Z，意思是X和Y之间的安全时间是Z，而且Y必须在X之后运行。这些指令从0到N-1标号。

**Output**

打印一个整数，CPU运行的最短时间

**Sample Input**

5 2

1 2 1

3 4 1

**Sample Output**

2

**HINT**

在第1ns中，指令0，1和3被执行了。

在第2ns中，指令2,和4被执行了

所以答案是2

**Poj3687 Labeling Balls**

**escription**

Windy 有N个不同重量的求，现在他打算给他们贴上1到N的标签：

没有两个相同的球

这些标签要符合规则例如“标签a球轻于标签b球”

你能帮他解决这个问题吗？

**Input**

第一行是数据个数. 每组数据第一行是两个整数： N (1 ≤ N ≤ 200) 和 M (0 ≤ M ≤ 40,000).接下来M行，每行两个整数a 和 b表示 标签a球轻于标签b球. (1 ≤ a, b ≤ N) 每组数据后有一个空行

**Output**

每组测试数据输出一行，按照小球的1~N，小球的重量输出，如果有多组解： you should output the one with the smallest weight for label 1, then with the smallest weight for label 2, then with the smallest weight for label 3 and so on... 如果无解，输出-1。

**Sample Input**

5

4 0

4 1

1 1

4 2

1 2

2 1

4 1

2 1

4 1

3 2

**Sample Output**

1 2 3 4

-1

-1

2 1 3 4

1 3 2 4