

# 树型动态规划七道题

## 一、选课

学校实行学分制。每门的必修课都有固定的学分，同时还必须获得相应的选修课程学分。学校开设了  $N$  ( $N < 300$ ) 门的选修课程，每个学生可选课程的数量  $M$  是给定的。学生选修了这  $M$  门课并考核通过就能获得相应的学分。

在选修课程中，有些课程可以直接选修，有些课程需要一定的基础知识，必须在选了其它的一些课程的基础上才能选修。例如《Frontpage》必须在选修了《Windows 操作基础》之后才能选修。我们称《Windows 操作基础》是《Frontpage》的先修课。每门课的直接先修课最多只有一门。两门课也可能存在相同的先修课。每门课都有一个课号，依次为 1, 2, 3, ...。例如：

课号	先修课号	学分
1	无	1
2	1	1
3	2	3
4	无	3
5	2	4

表中 1 是 2 的先修课，2 是 3、4 的先修课。如果要选 3，那么 1 和 2 都一定已被选修过。

你的任务是为自己确定一个选课方案，使得你能得到的学分最多，并且必须满足先修课优先的原则。假定课程之间不存在时间上的冲突。

**程序名：score**

**输入格式：**

输入文件的第一行包括两个整数  $N$ 、 $M$ （中间用一个空格隔开），其中  $1 \leq N \leq 300, 1 \leq M \leq N$ 。

以下  $N$  行每行代表一门课。课号依次为 1, 2, ...,  $N$ 。每行有两个数（用一个空格隔开），第一个数为这门课先修课的课号（若不存在先修课则该项为 0），第二个数为这门课的学分。学分是不超过 10 的正整数。

**输出格式：**

输出文件只有一个数：实际所选课程的学分总数。

**输入样例：**

```
7 4
2 2
0 1
0 4
2 1
7 1
7 6
2 2
```

**输出样例：**

```
13
```

**来源：**CTSC

## 二、战略游戏

Bob 喜欢玩电脑游戏，特别是战略游戏。但是他经常无法找到快速玩过游戏的办法。现在他有个问题。他要建立一个古城堡，城堡中的路形成一棵树。他要在这棵树的结点上放置

最少数目的士兵，使得这些士兵能了望到所有的路。注意，某个士兵在一个结点上时，与该结点相连的所有边将都可以被了望到。

请你编一程序，给定一树，帮 Bob 计算出他需要放置最少的士兵。

**程序名: stragedi**

**输入格式:**

输入文件中数据表示一棵树，描述如下:

第一行  $N$ ，表示树中结点的数目。

第二行至第  $N+1$  行，每行描述每个结点信息，依次为: 该结点标号  $i$ ,  $k$  (后面有  $k$  条边与结点  $i$  相连), 接下来  $k$  个数, 分别是每条边的另一个结点标号  $r_1, r_2, \dots, r_k$ 。

对于一个  $n$  ( $0 < n \leq 1500$ ) 个结点的树，结点标号在  $0$  到  $n-1$  之间，在输入文件中每条边只出现一次。

**输出格式:**

输出文件仅包含一个数，为所求的最少的士兵数目。

例如，对于如下图所示的树:



答案为 1 (只要一个士兵在结点 1 上)。

**输入样例 1:**

```
4
0 1 1
1 2 2 3
2 0
3 0
```

**输出样例 1:**

```
1
```

**输入样例 2:**

```
5
3 3 1 4 2
1 1 0
2 0
0 0
4 0
```

**输出样例 2:**

```
2
```

来源: SGOI

### 三、没有上司的晚会

Ural 大学有  $N$  个职员，编号为  $1 \sim N$ 。他们有从属关系，也就是说他们的关系就像一棵以校长为根的树，父结点就是子结点的直接上司。每个职员有一个快乐指数。现在有个周年庆典宴会，要求与会职员的快乐指数最大。但是，没有职员愿和直接上司一起与会。

**程序名: party**

**输入格式:**

第一行一个整数  $N$ 。 ( $1 \leq N \leq 6000$ )

接下来  $N$  行，第  $i+1$  行表示  $i$  号职员的快乐指数  $R_i$ 。 ( $-128 \leq R_i \leq 127$ )

接下来  $N-1$  行，每行输入一对整数  $L, K$ 。表示  $K$  是  $L$  的直接上司。

最后一行输入  $0, 0$ 。

**输出格式：**

输出最大的快乐指数。

**输入样例：**

```
7
1
1
1
1
1
1
1
1
1
1 3
2 3
6 4
7 4
4 5
3 5
0 0
```

**输入样例：**

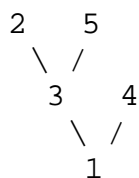
```
5
```

**来源：**URAL

## 四、二叉苹果树

有一棵苹果树，如果树枝有分叉，一定是分 2 叉（就是说没有只有 1 个儿子的结点）。这棵树共有  $N$  个结点（叶子点或者树枝分叉点），编号为  $1-N$ ，树根编号一定是 1。

我们用一根树枝两端连接的结点的编号来描述一根树枝的位置。下面是一颗有 4 个树枝的树：



现在这颗树枝条太多了，需要剪枝。但是一些树枝上长有苹果。

给定需要保留的树枝数量，求出最多能留住多少苹果。

**程序名：**apple

**输入格式：**

第 1 行 2 个数， $N$  和  $Q$  ( $1 \leq Q \leq N, 1 < N \leq 100$ )。

$N$  表示树的结点数， $Q$  表示要保留的树枝数量。接下来  $N-1$  行描述树枝的信息。

每行 3 个整数，前两个是它连接的结点的编号。第 3 个数是这根树枝上苹果的数量。

每根树枝上的苹果不超过 30000 个。

**输出格式：**

一个数，最多能留住的苹果的数量。

**输入样例：**

```
5 2
1 3 1
1 4 10
2 3 20
```

3 5 20

输入样例：

21

来源：URAL(广州六中信息学奥赛小组译)

## 五、比赛转播

一个电视网络计划转播一场重要的足球比赛。网络中的传输点和接收点（即用户）可以表示一棵树。这棵树的根是一个传输点，它将转播比赛。树的叶节点是可能要接受这场比赛的用户（他当然可以选择不看比赛，这样就不要交款）。其他非根节点，非叶节点的中间节点为数据的中转站。

将一个信号从一个传输点传到另一个传输点的花费是给定的。整个转播的费用就是每一个传输费用的总和。每一个用户（叶节点）都准备付一定的钱来看这场比赛。电视网络公司要决定是否要给这个用户提供电视信号。例如：给一个节点传输信息的花费太大，而他愿意的付款也很少时，网络公司可能选择不给他转播比赛。

写一个程序，找到一个传输方案使最多的用户能看到转播比赛，且转播的费用不超过所有接收信号用户的交款。

程序名：tele

输入格式：

输入文件的第一行包含两个整数  $N$  和  $M$  ( $2 \leq N \leq 3000, 1 \leq M \leq N-1$ )。  $N, M$  表示分别表示树的节点数和想观看比赛的用户数。树的根节点用 1 表示，中间节点的标号为  $2 \sim N-M$ ，用户的节点标号为  $N-M+1 \sim N$ 。

接下来的  $N-M$  行表示传输点的信息（依次是节点 1, 2……）：

$K \ A1 \ C1 \ A2 \ C2 \ \dots \ Ak \ Ck$

表示一个传输点将信号传给  $K$  个用户，每一个包含两个数  $A$  和  $C$ ， $A$  表示传输点或用户的节点号， $C$  表示传输的花费。

最后一行含有用户的数据，有  $M$  个整数表示他们看这场比赛愿意的付费。

输出格式：

仅一行包含一个整数，最大的用户数。

输入样例 1：

```
5 3
2 2 2 5 3
2 3 2 4 3
3 4 2
```

输出样例 1：

2

输入样例 2：

```
5 3
2 2 2 5 3
2 3 2 4 3
4 4 2
```

输出样例 2：

3

输入样例 3：

```
9 6
3 2 2 3 2 9 3
2 4 2 5 2
3 6 2 7 2 8 2
4 3 3 3 1 1
```

输入样例 3:

5

来源: 未知

## 六、偷天换日

神偷对艺术馆内的名画垂涎欲滴准备大捞一把。

艺术馆由若干个展览厅和若干条走廊组成。每一条走廊的尽头不是通向一个展览厅，就是分为两个走廊。每个展览厅内都有若干幅画，每副画都有一个价值。经过走廊和偷画都是要花费时间的。

警察会在  $n$  秒后到达进口，在不被逮捕的情况下你最多能得到的价值。

程序名: **steal**

输入格式:

第一行一个整数  $n(n \leq 600)$ 。

第二行若干组整数，对于每组整数  $(t, x)$ ， $t$  表示进入这个展览厅或经过走廊要耗费  $t$  秒的时间，若  $x > 0$  表示走廊通向的展览厅内有  $x$  幅画，接下来  $x$  对整数  $(w, c)$  表示偷一幅价值为  $w$  的画需要  $c$  秒的时间。若  $x = 0$  表示走廊一分为二。 $(t, c \leq 5; x \leq 30)$

输入是按深度优先给出的。房间和走廊数不超过 300 个。

输出格式:

仅一个整数，表示能获得的最大价值。

输入样例:

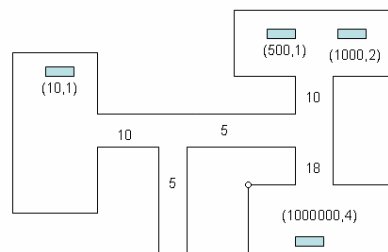
50

5 0 10 1 10 1 5 0 10 2 500 1 1000 2 18 1 1000000 4

输出样例:

1500

来源: 改编



## 七、河流

几乎整个 Byteland 王国都被森林和河流所覆盖。小点的河汇聚到一起，形成了稍大点的河。就这样，所有的河水都汇聚并流进了一条大河，最后这条大河流进了大海。这条大河的入海口处有一个村庄——名叫 Bytetown。

在 Byteland 国，有  $n$  个伐木的村庄，这些村庄都座落在河边。目前在 Bytetown，有一个巨大的伐木场，它处理着全国砍下的所有木料。木料被砍下后，顺着河流而被运到 Bytetown 的伐木场。Byteland 的国王决定，为了减少运输木料的费用，再额外地建造  $k$  个伐木场。这  $k$  个伐木场将被建在其他村庄里。这些伐木场建造后，木料就不用都被送到 Bytetown 了，它们可以在运输过程中第一个碰到的新伐木场被处理。显然，如果伐木场座落的那个村子就不用再付运送木料的费用了。它们可以直接被本村的伐木场处理。

注：所有的河流都不会分叉，形成一棵树，根结点是 bytetown。

国王的大臣计算出了每个村子每年要产多少木料，你的任务是决定在哪些村子建设伐木场能获得最小的运费。其中运费的计算方法为：每一吨木料每千米 1 分钱。

编一个程序:

1. 从文件读入村子的个数，另外要建设的伐木场的数目，每年每个村子产的木料的块数以及河流的描述。

2. 计算最小的运费并输出。

程序名: **river**

输入格式:

第一行 包括两个数  $n$  ( $2 \leq n \leq 100$ ),  $k$  ( $1 \leq k \leq 50$ , 且  $k \leq n$ )。  $n$  为村庄数,  $k$  为要建的伐木场的数目。除了 bytetown 外, 每个村子依次被命名为  $1, 2, 3, \dots, n$ , bytetown 被命名为  $0$ 。

接下来  $n$  行, 每行 3 个整数

$w_i$ ——每年  $i$  村子产的木料的块数 ( $0 \leq w_i \leq 10000$ )

$v_i$ ——离  $i$  村子下游最近的村子 (即  $i$  村子的父结点) ( $0 \leq v_i \leq n$ )

$d_i$ —— $v_i$  到  $i$  的距离 (千米)。(  $1 \leq d_i \leq 10000$  )

保证每年所有的木料流到 bytetown 的运费不超过  $2000,000,000$  分

50% 的数据中  $n$  不超过 20。

**输出格式:**

输出最小花费, 精确到分。

**输入样例:**

```
4 2
1 0 1
1 1 10
10 2 5
1 2 3
```

**输出样例:**

```
4
```

来源: IOI (Amber 译)

