网络流练习题

竞赛时间：2018 年 1 月 25 日 18:00－20:30

题目名称	染色	猜测	比特板
可执行文件名	paint	guess	bitboard
输入文件名	paint.in	guess.in	bitboard.in
输出文件名	paint.out	guess.out	bitboard.out
每个测试点时限	2 秒	1 秒	1 秒
内存限制	512MB	512MB	512MB
测试点数目	20	20	10
每个测试点分值	5	5	10
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型

染色

【问题描述】

有一棵n个点的有根树，节点的编号为0到$n-1$，其中根节点的编号为0。

有m种颜色。给一个点添上第i种颜色的代价为w_i。每个点必须被添上恰好两种不同的颜色。

定义一个点的控制集合为这个点的所有儿子与这个点自己构成的集合。注意如果b是a的儿子且c是b的儿子，c不是a的儿子。

我们称一个集合是多样的，当且仅当集合中的每个点的颜色都不同。这里一个点的颜色可以是其被添上的两种颜色的任意一种。

比如说，假设集合里有两个点，这两个点被添上的颜色都是(1,2)，那么集合是多样的，因为两个点的颜色可以是1和2。但是如果有集合里有三个点，这三个点被添上的颜色都是(1,2)，那么这个集合就不是多样的了。

称一个染色方案合法，当且仅当每个点的控制集合都是多样的。求一个染色代价最小的合法方案。如果不存在合法方案，则输出-1。

【输入格式】

输入文件的第一行包含两个整数n和m。

接下来一行包含$n-1$个整数，分别代表1到$n-1$号点的父亲。保证给定的父亲关系构成一棵树，而且i号点的父亲节点编号一定小于i。

接下来一行包含m个整数，第i个整数为w_i。

【输出格式】

输出一个整数，代表最小的染色代价。如果不存在合法方案，则输出-1。

【样例输入】

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

【样例输出】

```
10
```

【样例解释】

在最优方案中，0号节点和1号节点被添上的颜色是(1,2)，2号节点被添上的颜色是(1,3)。代价为$1+2+1+2+1+3=10$。

【数据规模和约定】

对于 10%的数据，$n,m \leq 5$。
对于 30%的数据，$m \leq 8$。
另外有 20%的数据，$w_2 = w_3 = \cdots = w_m$。
对于 100%的数据，$1 \leq n \leq 30$，$2 \leq m \leq 30$，$1 \leq w_i \leq 100$。

猜测

【问题描述】

有一块棋盘，棋盘的边长为 100000，行和列的编号为 1 到 100000。棋盘上有n个特殊格子，任意两个格子的位置都不相同。

现在小 K 要猜哪些格子是特殊格子。她知道所有格子的横坐标和纵坐标，但并不知道对应关系。换言之，她只有两个数组，一个存下了所有格子的横坐标，另一个存下了所有格子的纵坐标，而且两个数组都打乱了顺序。当然，小 K 猜的n个格子的位置也必须都不相同。

请求出一个最大的k，使得无论小 K 怎么猜，都能猜对至少k个格子的位置。

【输入格式】

输入数据第一行包含一个整数n。

接下来n行，每行描述一个特殊格子的位置。第i行含有两个整数x_i和y_i，代表第i个格子的坐标。保证任意两个格子的坐标都不相同。

【输出格式】

输出一行，包含一个整数，代表最大的k。

【样例输入 1】

```
2
1 1
2 2
```

【样例输出 1】

```
0
```

【样例解释 1】

小 K 有可能会猜$(1,2),(2,1)$，此时一个都没对。

【样例输入 2】

```
3
1 1
1 2
2 1
```

【样例输出 2】

```
3
```

【样例解释 2】

此时只有一种合法的猜测。注意$(1,1),(1,1),(2,2)$不是一个合法的猜测。

【数据规模和约定】

对于 30%的数据，$n \leq 8$。
另外有 5%的数据，所有横坐标和纵坐标均不相同。
另外有 15%的数据，所有横坐标或者纵坐标均不相同。
对于 100%的数据，$n \leq 50$，$1 \leq x_i, y_i \leq 100000$。

比特板

【问题描述】

天才发明家小 K 制造了一块比特板。板子上有2^n个比特元，编号为$0 \sim 2^n - 1$。每个比特元i可以接受一个$[0,2^m)$之间的整数作为输入信号，并产生整数p_i作为固定的输出信号。当编号为i的比特元接收到输入信号j时，将生成$W(i,j)$枚比特币。

相似的比特元之间还会产生叠加效果。每个比特元i有一个饱和值t_i。如果两个比特元的编号a和b在二进制下只有一位不同，并且两个比特元中的至少一个接收到的输入信号不小于其饱和之时，这两个比特元将额外生成$p_a \text{ xor } p_b$枚比特币。请注意，饱和值的取值范围是$[0,2^m]$。

小 K 希望给每个比特元适当的输入信号，使得比特版生成的比特币总数尽量多。小 K 觉得这个问题太简单了，于是他把问题交给你来解决。

【输入格式】

输入数据第一行包含两个整数n和m。

第二行包含2^n个整数t_i，代表每个比特元的饱和值。

第三行包含2^n个整数p_i，代表每个比特元的固定输出信号。

接下来2^n行，每行2^m个整数，代表$W(i,j)$。由于比特币是虚拟货币，所以$W(i,j)$可能是负数。

【输出格式】

输出一行，包含一个整数，代表最多能生成的比特币数。

【样例输入】

```
3 2
0 1 1 3 3 0 3 3
4 8 8 7 0 9 2 9
-9 -8 3 2
-9 -6 4 1
-6 -8 -5 3
3 -1 -4 -1
-6 -5 1 10
-10 7 3 -10
-3 -10 -4 -5
-2 -1 -9 1
```

【样例输出】

```
133
```

【样例解释】

输入信号依次为：2、2、3、0、3、1、0、3。

【数据规模和约定】

对于 20%的数据，$n \leq 3$，$m \leq 2$。
另外有 20%的数据，$t_i = 0$或2^m。
另外有 20%的数据，$m = 1$。
对于 100%的数据，$1 \leq n \leq 8$，$1 \leq m \leq 8$，$0 \leq t_i \leq 2^m$，$0 \leq p_i, |W(i,j)| \leq 2^{10}$。