

第 33 届全国信息学奥林匹克竞赛

CCF NOI 2016

湖南省省队集训模拟赛 by matthew99

竞赛时长：5 小时

题目名称	卡常大水题	经典傻逼题	辣鸡提答题
目录	shuiti	shabiti	tidati
可执行文件名	shuiti	shabiti	N/A
输入文件名	shuiti.in	shabiti.in	tidati*.in
输出文件名	shuiti.out	shabiti.out	tidati*.out
每个测试点时限	4 秒	1 秒	N/A
内存限制	512MB	512MB	N/A
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	否	否	是
题目类型	传统型	传统型	提交答案型
是否有附加文件	是	是	是

提交源程序须加后缀

对于 C++ 语言	shuiti.cpp	shabiti.cpp	N/A
对于 C 语言	shuiti.c	shabiti.c	N/A
对于 Pascal 语言	shuiti.pas	shabiti.pas	N/A

注意，最终测试时，所有编译命令均不打开任何优化开关。

卡常大水题

【问题描述】

这是一道卡常大水题，会做的人也不要激动，希望大家不要被卡常。

给定一张 n 个点的有向完全图，点的编号为1到 n ，每一条边有两个权值，注意 x 到 y 的边和 y 到 x 的边的权值不一定相同。

现在你想选出一些边，使得任意两个点都可以仅经过这些边互相到达，并且这个边集中每种权值的最大值的和尽量小。保证图中至少有两个点，因此该边集显然不能为空。

【输入格式】

从文件 `shuiti.in` 中读入数据。

输入的第一行包括一个整数 t ，表示数据组数。

每组数据的第一行包括一个整数 n ，表示有向完全图的点数。

接下来是一个 $n \times n$ 的矩阵，第 i 行第 j 列的整数 a_{ij} 当 $i = j$ 时是 0，当 $i \neq j$ 时表示点 i 与点 j 之间的连边的第一种权值。

接下来是一个 $n \times n$ 的矩阵，第 i 行第 j 列的整数 b_{ij} 当 $i = j$ 时是 0，当 $i \neq j$ 时表示点 i 与点 j 之间的连边的第二种权值。

【输出格式】

输出到 `shuiti.out` 中。

输出一行一个整数，表示最小的和。

【样例输入 1】

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

【样例输出 1】

```
3
```

【样例说明 1】

一种最优解是选择边 $1 \rightarrow 3$, $3 \rightarrow 2$, $2 \rightarrow 1$, 两种权值的最大值分别是 2, 1 因此答案是 $2 + 1 = 3$ 。

【样例输入输出 2】

见下发的 *shuiti/shuiti.in* 与 *shuiti/shuiti.ans*。

【数据规模和约定】

数据点	n 的规模	备注
1	$n \leq 5$	
2	$n \leq 50$	$a_{ij} = 0$
3	$n \leq 50$	$a_{ij} = a_{ji}, b_{ij} = b_{ji}$
4	$n \leq 50$	
5	$n \leq 50$	
6	$n \leq 50$	
7	$n \leq 150$	
8	$n \leq 150$	
9	$n \leq 150$	
10	$n \leq 150$	

对于所有数据, $2 \leq n \leq 150$, $0 \leq a_{ij}, b_{ij} \leq 10^9$, $a_{ii} = b_{ii} = 0$ 。

【提示】

本题时限较紧, 请注意尽量选择效率高的算法。

高维数组寻址的效率不高, 因此优化高维数组寻址可以获得很高的效率提升。

经典傻逼题

【问题描述】

这是一道经典傻逼题,对经典题很熟悉的人也不要激动,希望大家不要傻逼。

考虑一张 n 个点的带权无向图,点的编号为1到 n 。对于图中的任意一个点集(可以为空或者全集),所有恰好有一个端点在这个点集中的边组成的集合被称为割。一个割的权值被定义为所有在这个割上的边的异或和。

一开始这张图是空图,现在,考虑给这张无向图不断的加边,加入每条边之后,你都要求出当前权值最大的割的权值,注意加入的边永远都不会消失。

【输入格式】

从文件 *shabiti.in* 中读入数据。

输入的第一行包括一个数 id 表示数据编号,如第一组数据中的 $id = 1$ 。注意样例数据中的 $id = 0$ 。

接下来的第一行包括两个整数 n, m 表示图的点数和总共加的边。

接下来 m 行,每行三个正整数 x, y, w 表示在点 x 和点 y 之间加入一条权值为 w 的边。注意 x 和 y 可能相同,两条不同的边也可能连接了同一点。

此外, w 将以二进制形式从高位向低位给出,比如, $6 = 110_2$,因此如果边权为6,那么 w 将会是110。

【输出格式】

输出到 *shabiti.out* 中。

输出 m 行,按顺序输出每一条边加入之后权值最大的割的权值。

同样,你也要以二进制形式输出,形式和输入格式中描述的形式一样。

【样例输入 1】

```
0
3 6
1 2 1
1 2 1
3 3 111
1 3 101101
1 2 1011
2 3 111011
```

【样例输出 1】

```

1
0
0
101101
101101
110000

```

【样例说明 1】

前三条边加入之后的答案较为显然，考虑后三条边，加入第六条边之前，考虑点集 $\{1,2\}$ ，它对应的割只有第四条边，因此答案就是第四条边的权值，考虑加入最后一条边以后的情况，此时点集 $\{1,2\}$ 对应的割变成了第四条边和第六条边组成的集合，权值也发生了相应的改变。点集 $\{2\}$ 对应的割是第五条边和第六条边组成的集合，可以证明这就是权值最大的割，权值为 $1011_2 \oplus 111011_2 = 110000_2$ 。

【样例输入输出 2】

见下发的 *shabiti/shabiti1.in* 与 *shabiti/shabiti1.ans*。

【样例说明 2】

这组数据满足最终形成的图中每个点的度数不超过 1。

【样例输入输出 3】

见下发的 *shabiti/shabiti2.in* 与 *shabiti/shabiti2.ans*。

【数据规模和约定】

设 $l = \log_2 w$ 。

数据点	n 的规模	m 的规模	l 的规模	备注
1	$n \leq 20$	$m \leq 50$	$l < 32$	
2	$n \leq 20$	$m \leq 50$	$l < 32$	
3	$n \leq 200$	$m \leq 100$	$l < 100$	最终形成的图中每个点的度数不超过 1，且没有自环。
4	$n \leq 500$	$m \leq 250$	$l < 1000$	
5	$n \leq 500$	$m \leq 250$	$l < 1000$	
6	$n \leq 100$	$m \leq 200$	$l < 300$	

7	$n \leq 100$	$m \leq 200$	$l < 300$	
8	$n \leq 500$	$m \leq 1000$	$l < 1000$	
9	$n \leq 500$	$m \leq 1000$	$l < 1000$	
10	$n \leq 500$	$m \leq 1000$	$l < 1000$	

对于所有数据, $1 \leq n \leq 500$, $1 \leq m \leq 1000$, $0 \leq l < 1000$, $1 \leq x, y \leq n$ 。

辣鸡提答题

【问题描述】

这是一道辣鸡提答题，传统题不会做的人不要激动，希望大家不要全程做提答。

你有一个序列 $\{s_i\}$ ，满足 $s_1 = 1$ ，且对于所有 $x > 1$ ，存在 $y \leq x - 1$ ，使得 $s_x = s_{x-1} + s_y$ 。

给定 k ，现在要你构造一个长度为 n 的序列使得 $s_n = k$ ，并且 n 要求尽量小。

【输入格式】

所有输入数据 *tidati1.in~tidati10.in* 已在试题目录下。

一行一个正整数 k ，含义如题意中所述。

【输出格式】

针对给定的 10 个输入文件 *tidati1.in~tidati10.in*，你需要分别提交你的输出文件 *tidati1.out~tidati10.out*。

第一行输出 n ，表示序列的长度。

接下来一行 $n - 1$ 个正整数给出这个序列的生成规则，其中第 i 行的数 j 表示 $s_{i+1} = s_i + s_j$ 。请注意按照题意 $j \leq i$ 。

你输出的序列应满足 $s_n = k$ 。

【样例输入 1】

7

【样例输出 1】

7

1 1 1 1 1 1

【样例说明 1】

你也可以输出：

5

1 1 1 3

这个输出比样例输出要优。

【如何检测你的输出】

如果你的输出不合法，则直接判 0 分，所以请仔细检查输出是否合法。

在试题目录下给出了 10 个判分文件 *tidati1.scores~tidati10.scores*。每个判分文件均有 10 行 10 个单调递减的正整数，第 i 行的数表示如果要获得至少 i 分， n 至多可以是多少。

注意第一行的数不一定是最优解，只要你输出的 n 不超过这个数即可获得该测试点的满分，更优的答案也不能多得分。

【提示】

请妥善保存输入文件 *tidati*.in*，判分文件 *tidati*.scores* 和输出文件 *tidati*.out*，及时备份，以免误删。