

比较锻炼思维的题目，不限定类型

Translate:USACO/shuttle

Shuttle Puzzle 棋盘游戏

传统题目

译 by Jeru

描述

大小为 3 的棋盘游戏里有 3 个白色棋子，3 个黑色棋子，和一个有 7 个格子一线排开的木盒子。3 个白棋子被放在一头，3 个黑棋子被放在另一头，中间的格子空着。

初始状态: WWW_BBB
目标状态: BBB_WWW

在这个游戏里有两种移动方法是允许的：

1. 你可以把一个棋子移到与它相邻的空格；
2. 你可以把一个棋子跳过一个(仅一个)与它不同色的棋子到达空格。

大小为 N 的棋盘游戏包括 N 个白棋子， N 个黑棋子，还有有 $2N+1$ 个格子的木盒子。

这里是 3-棋盘游戏的解，包括初始状态，中间状态和目标状态：

WWW BBB
WW WBBB
WWBW BB
WWBWB B

```
WWB BWB
W BWBWB
WBWBWB
```

```
BW WBWB
BWBW WB
BWBWBW
BWBWB W
BWB BWB
B BWBWW
BB WBWW
BBBW WW
BBB WWW
```

请编一个程序解大小为 N 的棋盘游戏($1 \leq N \leq 12$)。要求用最少的移动步数实现。

格式

PROGRAM NAME: shuttle

INPUT FORMAT:

(file shuttle.in)

一个整数 N 。

OUTPUT FORMAT:

(file shuttle.out)

用空格在棋盘的位置(位置从左到右依次为 $1, 2, \dots, 2N+1$)表示棋盘的状态。

输出棋盘的状态变换序列，每行 20 个数(除了最后一行)。

输出的解还应当有最小的字典顺序(即如果有多组移动步数最小的解，输出第一个数最小的解；如果还有多组，输出第二个数最小的解；...)。

SAMPLE INPUT

3

SAMPLE OUTPUT

3 5 6 4 2 1 3 5 7 6 4 2 3 5 4

分析:

先观察样例数据，如果把还没移动的那一步也算上，那么空格的位置为

4 3 5 6 4 2 1 3 5 7 6 4 2 3 5 4 (n=3,样例)

5 4 6 7 5 3 2 4 6 8 9 7 5 3 1 2 4 6 8 7 5 3 4 6 5 (n=4)

我们凭借极其敏锐的眼光发现这组序列为

4 35 642 1357 642 35 4 (n=3,样例)

5 46 753 2468 97531 2468 753 46 5 (n=4)

即长度为 $1, 2, 3, 4, \dots, n, n+1, n, \dots, 4, 3, 2, 1$ 这样的 $2n+1$ 组等差序列

我们讨论第 $1 \sim n+1$ 组序列，这些序列满足

- *公差的绝对值为 2

- *奇数组为降序列，偶数组为升序列

- *对于第 i 组 ($1 \leq i \leq n+1$), 若为奇数组则首项为 $n+i$, 偶数组则首项为 $n-i+2$

对于第 $n+2 \sim 2n+1$ 组，可以由对称性求出。

输出时从第二组开始即可。

Translate:USACO/milk6

Pollutant Control 追查坏牛奶

译 by Twink

描述

你第一天接手三鹿牛奶公司就发生了一件倒霉的事情：公司不小心发送了一批有三聚氰胺的牛奶。很不幸，你发现这件事的时候，有三聚氰胺的牛奶已

经进入了送货网。这个送货网很大，而且关系复杂。你知道这批牛奶要发给哪个零售商，但是要把这批牛奶送到他手中有许多种途径。送货网由一些仓库和运输卡车组成，每辆卡车都在各自固定的两个仓库之间单向运输牛奶。在追查这些有三聚氰胺的牛奶的时候，有必要保证它不被送到零售商手里，所以必须使某些运输卡车停止运输，但是停止每辆卡车都会有一定的经济损失。你的任务是，在保证坏牛奶不送到零售商的前提下，制定出停止卡车运输的方案，使损失最小。

{批注：先前不知是哪个写了一个要不得的翻译，这才是正常的翻译，读起来也觉得比较健康——Cow-Tsc} {修改下，更现实——pyh119}

格式

PROGRAM NAME: milk6

INPUT FORMAT:

(file milk6.in) 第一行：两个整数 $N(2 \leq N \leq 32)$ 、 $M(0 \leq M \leq 1000)$ ， N 表示仓库的数目， M 表示运输卡车的数量。仓库 1 代表发货工厂，仓库 N 代表有三聚氰胺的牛奶要发往的零售商。第 $2..M+1$ 行：每行 3 个整数 S_i, E_i, C_i 。其中 S_i, E_i 表示这辆卡车的出发仓库，目的仓库。 $C_i(0 \leq C_i \leq 2,000,000)$ 表示让这辆卡车停止运输的损失。

OUTPUT FORMAT:

(file milk6.out) 第 1 行两个整数 c, t ， c 表示最小的损失， T 表示要停止的最少卡车数。接下来 t 行表示你要停止哪几条线路。如果有多种方案使损失最小，

输出停止的线路最少的方案。如果仍然还有相同的方案，请选择开始输入顺序最小的。

SAMPLE INPUT

```
4 5
1 3 100
3 2 50
2 4 60
1 2 40
2 3 80
```

SAMPLE OUTPUT

```
60 1
3
```

分析：

求最小割方案。我的方法是：根据最大流最小割定理，满足最大流的最小割的边集一定是饱和边。那么删除一条路径，再次查找最大流，如果与先前的差值恰好为删除路径的容量，则此路径包含于最小割集合中。查找时，分为两类：一类是最小割是一条割边，这样直接按照顺序遍历即可。另一类是最小割多于一条，同样按顺序遍历，保证题目要求。

Translate:USACO/frameup

Frame Up 重叠的图像

译 by Felicia Crazy

描述

看下面的五张 9×8 的图像：

```
..... .CCC....
```

EEEEEE..BBBB..	.C.C....
E...E..	DDDDDD..B..B..	.C.C....
E...E..	D...D..B..B..	.CCC....
E...E..	D...D..AAAA	..B..B..
E...E..	D...D..A..A	..BBBB..
E...E..	DDDDDD..A..A
E...E..AAAA
EEEEEE..

1	2	3	4	5
---	---	---	---	---

现在，把这些图像按照 1—5 的编号从下到上重叠，第 1 张在最下面，第 5 张在最顶端。如果一张图像覆盖了另外一张图像，那么底下的图像的一部分就变得不可见了。我们得到下面的图像：

	.CCC....
	ECBCBB..
	DCBCDB..
	DCCC.B..
	D.B.ABAA
	D.BBBB.A
	DDDDAD.A
	E...AAAA
	EEEEEE..

对于这样一张图像，计算构成这张图像的矩形图像从底部到顶端堆叠的顺序。

下面是这道题目的规则：

- 矩形的边的宽度为 1，每条边的长度都不小于 3。
- 矩形的每条边中，至少有一部分是可见的。注意，一个角同时属于两条边。
- 矩形用大写字母表示，并且每个矩形的表示符号都不相同。

格式

PROGRAM NAME: frameup

INPUT FORMAT:

(file frameup.in)

第一行 两个用空格分开的整数: 图像高 H ($3 \leq H \leq 30$) 和图像宽 W ($3 \leq W \leq 30$) 。

第二行到第 $H+1$ 行 H 行, 每行 W 个字母。

OUTPUT FORMAT:

(file frameup.out)

按照自底向上的顺序输出字母。如果有不止一种情况, 按照字典顺序输出每一种情况 (至少会有一种合法的顺序)。

SAMPLE INPUT

```
9 8
. CCC. . . .
ECBCBB. .
DCBCDB. .
DCCC. B. .
D. B. ABAA
D. BBBB. A
DDDDAD. A
E. . . AAAA
EEEEEE. .
```

SAMPLE OUTPUT

```
EDABC
```

分析:

本题目具有明显的层次特征, 可以使用拓扑排序, 而且一定要理解题意, 题目的“矩形的每条边中, 至少有一部分是可见的”这句话, 意思是每个矩形的四边都有可见部分, 这样我们就能计算出一个矩形的尺寸, 然后在他的每条边上扫描, 只要有不属于矩形的边存在, 矩形就被那个矩形压着, 连一条有向边。这样进行排序就构成了拓扑序列。不过题目要求输出所有的拓扑序, 因而一般的 BFS 找到一个解是不行的。比较直观的是 DFS 所有可行的解, 然后

排序。

DFS 求出所有可行解的方法是：

每次都要遍历**所有**矩形，并选出一个入度为零的矩形，同时删除矩形相连的所有边。并给当前的被选的矩形标号（保证以后不会选重）。直到达到限定深度，层次的限定由矩形的个数决定。这样往复，找到所有解。