本章为图论初步和种子染色法(附带其他)
Translate: USACO/castle

The Castle 城堡

描述

我们憨厚的 USACO 主人公农夫约翰(Farmer John)以无法想象的运气,在他生日那天收到了一份特别的礼物：一张“幸运爱尔兰”（一种彩票）。结果这张彩票让他获得了这次比赛唯一的奖品——坐落于爱尔兰郊外的一座梦幻般的城堡！(RP...)

喜欢吹嘘的农夫约翰立刻回到有着吹嘘传统的威斯康辛老家开始吹嘘了，农夫约翰想要告诉他的奶牛们关于他城堡的一切。他需要做一些吹嘘前的准备工作：比如说知道城堡有多少个房间，每个房间有多大。另外，农夫约翰想要把一面单独的墙（指两个单位间的墙）拆掉以形成一个更大的房间。你的工作就是帮农夫约翰做以上的准备，算出房间数与房间的大小。

城堡的平面图被划分成 M*N（M 是宽度）个正方形的单位，一个这样的单位可以有 0 到 4 面墙环绕。城堡周围一定有外墙环绕以遮风挡雨。（就是说平面图的四周一定是墙。）

请仔细研究下面这个有注解的城堡平面图：

```
      1   2   3   4   5   6   7
    #############################
 1  #   |   #   |   #   |   |   #
    #####---#####---#---#####---#
 2  #   #   |   #   #   #   #   #
    #---#####---#####---#####---#
 3  #   |   |   #   #   #   #   #
    #---#########---#####---#---#
 4  # ->#   |   |   |   |   #   #
    #############################
# =墙壁      -,| = 没有墙壁
-> =指向一面墙，这面墙推掉的话我们就有一间最大的新房间
```

友情提示，这个城堡的平面图是 7×4 个单位的。一个“房间”指的是平面图中一个连通的“正方形单位”的集合。比如说这个样例就有 5 个房间。（大小分别为 9、7、3、1、8 个单位（排名不分先后））

移去箭头所指的那面墙，可以使 2 个房间合为一个“移掉一面单独的墙可以形成的最大的新房间”。（原文为：Removing the wall marked by the arrow merges a pair of rooms to make the largest possible room that can be made by removing a single wall. ）

城堡保证至少有 2 个房间，而且一定有一面墙可以被移走。

格式

PROGRAM NAME: castle

INPUT FORMAT: 城堡的平面图用一个由数字组成的矩阵表示，一个数字表示一个单位，矩阵有 N 行 M 列。输入与样例的图一致。

每一个单位的数字告诉我们这个单位的东西南北是否有墙存在。每个数字是由以下四个整数的某个或某几个或一个都没有加起来的。

```
1：在西面有墙
2：在北面有墙
4：在东面有墙
8：在南面有墙
```

城堡内部的墙会被规定两次。比如说（1，1）南面的墙，亦会被标记为（2，1）北面的墙。

(file castle.in)

第 1 行: 二个被空格分开的整数: M 和 N（N,M<=50）

第 2 到 N+1 行: M x N 个整数,每行 M 个。

OUTPUT FORMAT:

(file castle.out)

输出包含如下 4 行:

第 1 行: 城堡的房间数目。

第 2 行: 最大的房间的大小

第 3 行: 移除一面墙能得到的最大的房间的大小

第 4 行: 移除哪面墙

要造出第三行那么大的大房间所要推倒的一面墙。

选择最佳的墙来推倒。有多解时选最西边的（仍然有多解时选这些里面最南的）。用这面墙南边或西边的单位，还有这面墙在那个单位的方位（"N"（北）或者"E"（东））来表示这面墙。

SAMPLE INPUT

```
7 4
11 6 11 6 3 10 6
7 9 6 13 5 15 5
1 10 12 7 13 7 5
13 11 10 8 10 12 13
```

SAMPLE OUTPUT

```
5
9
16
4 1 E
```

题解 - castle

分析

`floodfill`

用一个二维数组记录每个格的四面是否有墙，根据输入，从最大的 8 到最小的 1 一次减直到减到 0，得出每个格的墙的情况。 然后用 `floodfill`，对每个房间染色，求出房间数，和每个房间的面积，输出其中最大的。

枚举每堵墙，如果墙的两边不是同一个房间且其面积和更大，更新结果，为了结果唯一，我们从左下向右上，一列一列枚举格子，并且先枚举该格上面的墙再枚举右边的。

并查集

也是可以的，判断墙的状况可以用位运算

Translate:USACO/frac1

Ordered Fractions 顺序的分数

描述

输入一个自然数 N,请写一个程序来增序输出分母小于等于 N 的最简既约分数

格式

PROGRAM NAME: frac1

INPUT FORMAT:

(file frac1.in)

单独的一行　一个自然数 N(1..160)

OUTPUT FORMAT:

(file frac1.out)

每个分数单独占一行，按照大小次序排列

SAMPLE INPUT

```
5
```

SAMPLE OUTPUT

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

解题报告:

快排

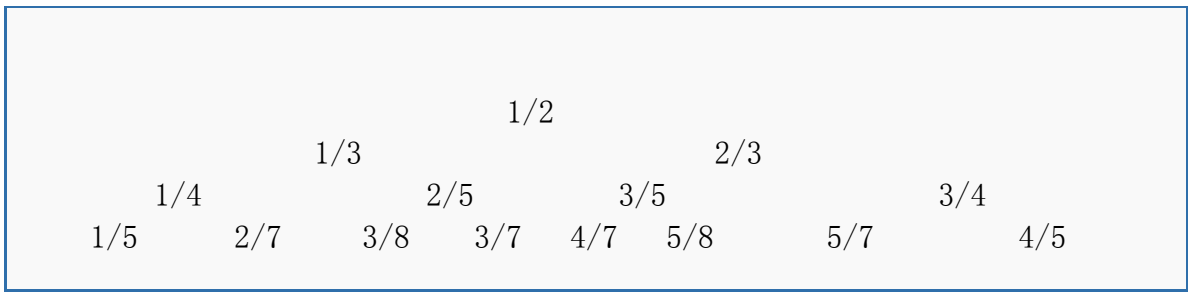
枚举所有的分数，判断其是否是最简（分母分子最大公约数＝1），用一个数列记录所有最简分数，然后用快排排序。

数学

A better solution from Russ:

We notice that we can start with 0/1 and 1/1 as our ``endpoints *and recursively generate the middle points by adding numerators and denominators.*

0/1 1/1



Each fraction is created from the one up to its right and the one up to its left. This idea lends itself easily to a recursion that we cut off when we go too deep.

Translate:USACO/sort3

Sorting a Three-Valued Sequence 三值的排序

IOI'96 - Day 2

描述

排序是一种很频繁的计算任务。现在考虑最多只有三值的排序问题。一个实际的例子是，当我们给某项竞赛的优胜者按金银铜牌序的时候。在这个任务中可能的值只有三种 1，2 和 3。我们用交换的方法把他排成升序的。

写一个程序计算出，给定的一个 1,2,3 组成的数字序列，排成升序所需的最少交换次数。

格式

PROGRAM NAME: sort3

INPUT FORMAT:

(file sort3.in)

Line 1:

N (1 <= N <= 1000)

Lines 2-N+1:

每行一个数字，共 N 行。（1..3）

OUTPUT FORMAT:

(file sort3.out)

共一行，一个数字。表示排成升序所需的最少交换次数。

SAMPLE INPUT

```
9
2
2
1
3
3
3
2
3
1
```

SAMPLE OUTPUT

```
4
```

题解 - sort3

分析

Way1:

用 l[i]记录 i(1,2,3)出现的个数。排序后一定是 l[1]个 1，l[2]个 2，l[3]个 3 这 3 段。

sn[i,j]记录在第 i 段中的 j 的个数。第 i 段中的 j 和第 j 段中的 i 交换，两个元素交换，只需要一次。所以取 sn[i,j]和 sn[j,i]中的较小数 s，累加到总交换数中。

这样交换后，剩下的肯定是 3 段同时交换，最少需要 2 次。我们只需累加(sn[1,2]+sn[1,3])×2 即可。

Way2

这个题目的思路很简单:首先读入所有数,然后看原本的位置是’1′的里面有几个不是 1,如果那几个不是 1 的数字在它原应该呆的位置中找到 1 的话就直接换,如果找不到就在另一堆中找…

比如说 3 1 2 三个数,第一个位置原应该是 1,但是是 3,那么就在原应该是 3 的位置中找 1,但是是 2,那么就在另一堆找,找到了 1,那么把 3 跟 1 换一下…直到 1 全部到达自己的位置…然后在原应该是 2 的位置中找,如果碰到 3 就应该换一次….就这样找完就行了…

```
for i:=1 to n do
begin
  readln(a[i]);
  if a[i]=1 then inc(s[1,2])
  else
  if a[i]=2 then inc(s[2,2]);
end;
s[2,1]:=s[1,2]+1;
s[2,2]:=s[2,2]+s[2,1]-1;
s[3,1]:=s[2,2]+1;
```

这段代码是记录位置的,s[x,1]是 x 的初始位置… s[x,2]是 x 的终点位置…

记录好位置之后就按照上面的流程去找就好了…实现起来应该是很简单的…

Translate:USACO/holstein

Healthy Holsteins 健康的荷斯坦奶牛

描述

农民 JOHN 以拥有世界上最健康的奶牛为骄傲。他知道每种饲料中所包含的牛所需的最低的维他命量是多少。请你帮助农夫喂养他的牛，以保持它们的健康，使喂给牛的饲料的种数最少。

给出牛所需的最低的维他命量，输出喂给牛需要哪些种类的饲料，且所需的饲料剂量最少。

维他命量以整数表示，每种饲料最多只能对牛使用一次，数据保证存在解。

格式

PROGRAM NAME: holstein

INPUT FORMAT:

(file holstein.in)

第 1 行：一个整数 V(1<=V<=25)，表示需要的维他命的种类数。

第 2 行：V 个整数(1<=每个数<=1000)，表示牛每天需要的维他命的最小量。

第 3 行：一个整数 G(1<=G<=15)，表示可用来喂牛的饲料的种数。

下面 G 行，第 i 行表示编号为 i 饲料包含的各种维他命的量的多少。

OUTPUT FORMAT:

(file holstein.out)

输出文件只有一行，包括：

牛必需的最小的饲料种数 P

后面有 P 个数，表示所选择的饲料编号（按从小到大排列）。

如果有多个解，输出饲料种类最小的。

SAMPLE INPUT

```
4
100 200 300 400
3
50 50 50 50
200 300 200 300
900 150 389 399
```

SAMPLE OUTPUT

```
2 1 3
```

Translate:USACO/hamming

Hamming Codes 海明码

描述

给出 N，B 和 D：找出 N 个编码（1 <= N <= 64）（实际评测中 2 <= N <= 60），每个编码有 B 位[二进制]（1 <= B <= 8），使得两两编码之间至少有 D 个单位的“海明距离”（1 <= D <= 7）。“海明距离”是指对于两个编码，他们的二进制表示法中的不同二进制位的数目。看下面的两个编码 0x554 和 0x234 之间的区别（0x554 表示一个十六进制数，每个位上分别是 5，5，4）：

```
0x554 = 0101 0101 0100
0x234 = 0010 0011 0100(不是将每一位分别转为 2 进制，这里空格只是为了方便演示)
不同位     xxx   xx
```

因为有五个位不同，所以“海明距离”是 5。

格式

PROGRAM NAME: hamming

INPUT FORMAT:

(file hamming.in)

一行，包括 N, B, D。

OUTPUT FORMAT:

(file hamming.out)

N 个编码（用十进制表示），要排序，十个一行。如果有多解，你的程序要输出这样的解：假如把它化为 2^B 进制的数，它的值要最小。

SAMPLE INPUT

```
16 7 3
```

SAMPLE OUTPUT

```
0 7 25 30 42 45 51 52 75 76
82 85 97 102 120 127
```

提示

必须与其他所有的数相比、海明码都符合要求,这个数才正确!