**题目1：无尽的编号**

时间限制: 1000ms 内存限制: 256MB

**描述**

在一条公路上，将要依次建造N座建筑。在每个建筑建成之后，都会用一个01串来给它编号。整条公路从起点到终点，所有建筑的编号都严格按照字典序递增的顺序来排列，而每在一个新的地方建起一个建筑时，它的编号会按以下规则确定：

1) 编号要比前一个建筑**（起点方向）**的字典序大，比后一个建筑**（终点方向）**的字典序小

3) 编号一定以1结尾

2) 编号要尽可能短，满足该条件时，字典序尽可能小

最开始时，公路的起点和终点上各有一个建筑，编号分别是0和1。接下来依次给出N个坐标 a1, a2, ..., aN，依次表示下一个建筑将要建造的位置，最后要问，当所有建筑建成时，这些建筑的编号总长度是多少，其中又出现了多少个字符1。所有建筑都在公路起点和终点之间，并且没有两个建筑建在同一个位置。

**输入**

输入文件包含多组测试数据。

第一行，给出一个整数T，为数据组数。接下来依次给出每组测试数据。

每组数据中第一行为一个整数 N，表示将要建造的建筑数量，第二行是用单个空格隔开的N个互不相同的整数 a1, a2, ..., aN，表示依次将要建造的建筑所在的坐标。

**输出**

对于每组测试数据，输出一行"Case #X: Y Z"，其中X表示测试数据编号，Y表示所有建筑编号总长，Z表示所有编号中字符1的数量。所有建筑包括起点和终点的这两个建筑。所有数据按读入顺序从1开始编号。

**数据范围**

小数据：T ≤ 100, 0 < N ≤ 100, 0 ≤ ai ≤ 1000

大数据：T ≤ 10, 0 < N ≤ 50000, 0 ≤ ai ≤ 500000

**样例输入**

1

5

1 2 3 4 5

**样例输出**

Case #1: 22 16

**解题思路**

我们简单画几个样例来模拟一下规则，就可以看出其中的规律了。首先，考虑新建筑的编号的长度，它一定比相邻两个编号更长，再根据编号规则可以知道，它的长度一定是相邻编号长度较大值+1。其次，考虑新编号中1的个数，它总是恰好等于左边建筑的1的数量+1。找到这两个规律之后，我们便可以通过模拟来得到每个建筑的编号长度和其中1的个数。在模拟中，需要用到线段树或者类似的数据结构，来实时查询每个新建筑位置的相邻两个建筑是什么。

**题目2：招聘**

时间限制: 3000ms 内存限制: 256MB

**描述**

Alice新开了一家公司，它的下面有两个项目，分别需要N1和N2个人来完成。现在有N个人前来应聘，于是Alice通过面试来决定他们中的哪些人会被录用。

Alice在面试中，会仔细考察他们能如何为公司的项目带来收益。她给每个人打了两个分值Q1和Q2，表示他加入第一个和第二项目分别能带来的收益值。同时，她也会仔细考察他们每个人的缺点，并且给每人打了另两个分值C1和C2，表示他们进入每个项目可能带来的负面效应。Alice心目中的最优决策是，在决定好录用哪些人以及每个人在哪个项目下工作之后，他们为公司带来的收益总和，除以他们为项目带来的负面效应总和，这个比值要最大。你能帮他计算出在最优决策下，这个比值为多少吗？

前来应聘的人数总是大于等于两个项目需求人数的总和，因此Alice一定会恰好招N1+N2个人，分配给第一个项目N1个人，分配给第二个项目N2个人，没有人会同时属于两个项目。

**输入**

输入文件包含多组测试数据。

第一行，给出一个整数T，为数据组数。接下来依次给出每组测试数据。

每组数据第一行为三个用空格隔开的整数N，N1，N2，表示前来应聘的人数，以及两个项目分别需要的人数。

接下来N行，每行是用空格隔开的四个整数Q1，C1，Q2，C2，依次表示每个人在第一个项目下的价值和负面效应，以及第二个项目下的价值和负面效应。

**输出**

对于每组测试数据，输出一行"Case #X: Y"，其中X表示测试数据编号，Y表示最优决策下招募的人的价值总和与负面效应总和的比值，与正确答案的绝对误差不应超过10-6。所有数据按读入顺序从1开始编号。

**数据范围**

T ≤ 100

1 ≤ Q1, Q2 ≤ 2000

1 ≤ C1, C2 ≤ 50

小数据：0 < N1 + N2 ≤ N ≤ 50,

大数据：0 < N1 + N2 ≤ N ≤ 500

**样例输入**

1

5 2 2

12 5 8 3

9 4 9 4

7 3 16 6

11 5 7 5

18 10 6 3

**样例输出**

Case #1: 2.444444

**解题思路**

这是一道典型的01分数规划问题，我们想要求这样一个函数的最大值 f(x)/g(x)（g(x)>0）, 我们只需找到最小的r，使得存在x满足h(x)=f(x)-r\*g(x)>=0，这个r即为解。而这个r可以通过二分来确定。我们现在只需在给定r的条件下，求解f(x)-r\*g(x)的最大值即可。

在这道问题中，f(x)=sum(Q1[i])+sum(Q2[j]) (i in S1, j in S2), g(x)=sum(C1[i])+sum(C2[j]) (i in S1, j in S2), 我们需要确定出两个集合S1,S2，使得f(x)-r\*g(x)最大，即h(x)=sum(Q1[i]-r\*C1[i])+sum(Q2[j]-r\*C2[j]) (i in S1, j in S2)。这里我们只需用动态规划来求解。令f[i][j][k]表示，在前i个人之中，有j个人属于S1，有k个人属于S2，此时最大的h(x)函数值为多少，转移很好考虑，即下一个人分在哪个项目或者不招进来。这样便能找到最大的h(x)。

仅仅做到这里还是不够，考虑到数据范围，这种做法可能会超时。一种优化思路是，当S1集合的人定下来时时，S2集合中的人一定可以贪心地选择剩余人中Q2[j]-r\*C2[j]值最大的前N2个人，由此我们修改这个算法，先将所有人按Q2[j]-r\*C2[j]从大到小排序，然后令f[i][j]表示前i个人之中，有j个人属于S2，并且不属于S1的人自动选前N2个人属于S2，此时h(x)的最大值。转移方式类似。如此一来，时间复杂度会下降一个阶，可以解决这道题目。

**题目3：游泳池**

时间限制: 6000ms 内存限制: 256MB

**描述**

Alice要在家里后院修建游泳池用来开party。她考察了一下，发现在游泳池开party大家最喜欢去的地方其实就是水边。人们既可以在泡在水里坐在边上的台阶上，也可以在岸上坐着晒太阳，总之就是没有人会真正在里面游泳的。所以Alice比较关心自己家修建的游泳池的周长。

Alice家的后院是一个矩形，有些地方已经布好了水龙头，有些地方种着不方便移动的大树。Alice想要建造一个或多个游泳池，使得周长正好等于S。当然了，每个游泳池至少要覆盖了一个水龙头，也不能砍掉任何一颗树。游泳池可以是任意形状。所以一个合法的游泳池应该是一片连通的、包含至少一个水龙头且不包含任何树的区域。

你可以帮Alice算一下，不同周长下各有多少种方案么？

**输入**

输入数据的第一行包含一个整数T，表示数据组数。

接下来有T组数据，每组数据中：

第一行包含两个整数N,M，表示后院的大小。

接下来N行M列的矩阵表示后院的布局。其中'.'表示空地，'T'表示大树，'X'表示水龙头。

**输出**

对于每组数据，先输出一行"Case #X:",其中X为数据组数编号，从1开始。

接下来对于可能出现的周长，输出一行"Len Count"，表示周长为Len的方案数共有Count个，因为Count可能很大，所以只需输出Count mod 100000007 (108 + 7)。按周长从小到大排序。

**数据范围**

1 ≤ T ≤ 20

小数据：1 ≤ N, M ≤ 6

大数据：1 ≤ N, M ≤ 8

**样例输入**

2

2 2

..

TX

2 3

XT.

.TX

**样例输出**

Case #1:

4 1

6 1

8 1

Case #2:

4 2

6 2

8 1

10 2

12 1

**解题思路**

此题明显是连通性状态压缩动态规划的题目，算法相对也比较明了，难点在于细节和代码量以及调试上。状态一定是需要用到连通块最小表示的。

标程中状态的设计为：

0表示不被变成游泳池的地方

1表示要变成游泳池且已经连通了水龙头的地方

2表示要变成游泳池且未连通水龙头且在当前边界上没有其他格子与其相连通的地方

3,4,5...表示要变成游泳池且未连接水龙头的大小大于1的连通分量

转移时即考虑当前格是否要建成游泳池，然后相应的计算周长，调整状态（比如合并连通分量，调整连通到水龙头的状态等等，这里就不一一列举了）。最终将所有的合法状态统计一遍即可。

**题目4：折手帕**

时间限制: 1000ms 内存限制: 256MB

**描述**

你打算送给女神一件有些羞羞的礼物：一块特殊的手帕。这块手帕是个四边形，但不像通常的手帕那样是一个正方形，手帕四条边的长度有着特殊的含义。例如你送给女神的手帕四边长分别是 5, 20, 13, 14，多么隐晦的表白呀！此外，这块手帕还是个凸的四边形。一个多边形被称为凸的是指，任意一条两端点在多边形内的直线段都完全包含在多边形内。你认为这寓意着“幸福美满”。

现在你正准备包装这件礼物。为了节省包装纸，你想通过折叠手帕来减少面积。同时，为了不让手帕过于褶皱，你只能折叠一次。此外，你还希望折叠后的手帕仍然是个凸的多边形。你想知道怎样折叠才能满足上述的要求，并且让折叠后的手帕面积最小。

**输入**

输入包括多组数据。第一行是整数 T，表示有多少组测试数据。接下来每行包含 5 个正整数，L1, L2, ..., L5。如果记四边形为ABCD，那么这 5 个整数分别是边 AB, BC, CD, AD, AC 的长度。输入数据保证通过这 5 个数可以唯一确定一个不退化的四边形。

**输出**

对于每组测试数据，单独输出一行“Case #c: a”。其中，c 表示测试数据的编号（从 1 开始），a 表示折叠后手帕的最小面积，a 与最优解的绝对误差在10-4以内即可。

**数据范围**

1 ≤ T ≤ 100

1 ≤ Li ≤ 1000

小数据：L1 = L3, L2 = L4

大数据：不存在上述条件

**样例输入**

4

4 3 5 6 5

25 15 7 15 20

3 4 3 4 5

10 10 10 10 14

**样例输出**

Case #1: 10.5000

Case #2: 96.0000

Case #3: 6.0000

Case #4: 49.9900

**解题思路**

此题要求折叠一个凸四边形，使得新的多边形仍然为凸的，求最小的可能面积。

首先，为了让面积最小，四边形的某一点必然要和某一边接触。假设沿直线l对折四边形ABCD，l把ABCD分为S1和S2两部分，接下来分情况讨论：

1. S1和S2互不为子集：

要使得新多边形为凸的，只能有两种情况：某一顶点与另一顶点重合，或某一边与另一边共线。具体来说，不失一般性地，有四种情况：(a)点A与点B重合，(b)点A与点C重合，(c)边AB与边AD共线，(d)边AB与边CD共线。其他情况可以通过对四边形的旋转和翻转获得。

2. S1为S2的子集（S2为S1的子集也类似）：

此时S1完全被S2覆盖，不妨设S1中的点A接触到了S2的某一边。想象A被“吸附”在了S2的边沿，并沿着边沿移动，如果A没有接触到S2的顶点，折痕l也没有接触ABCD的顶点，那么在每段连续的区间上，新多边形的面积是单调变化的，所以我们只需要枚举A接触S2的顶点，或l接触ABCD的顶点的情况。具体来说，除去情况1中的(a)和(b)，还有三种情况：(e)l为对角线BD，(f)l经过点A，而B恰好接触到了CD，(g)l为BC的垂线，A恰好接触到了CD

枚举上述七种情况时，还需要判断新的多边形是否确实是凸的。此外还要对ABCD进行旋转和翻转。