

HDU-multi-6 by FZDX

HDU-multi-6 by FZDX

Problem List

Tutorial

1001 oval and rectangle

1002 bookshelf

Part 1

Part 2

Part 3

1003 ring land

Part 0

Part 1

Part 2

Part 3

Part 4

1004 Shoot Game

1005 black and white

Part 1

Part 2

Part 2

Part 3

1006 foam transformation

Part 1

Part 2

Part 3

Part 4

1007 Variance-MST

1008 Rectangle Outline

1009 Werewolf

Part 1

1010 Chopping hands

Part 1

Part 2

Part 3

后记

1011 sacul

part 1

part 2

part 3

1012 Pinball

Part 1

Part 2

Problem List

- 难度分布: 签到 × 1, 前期 × 2, 中期 × 4, 中后 × 1, 后期 × 2, 防穿 × 2.

id	tag	预计/实际通过
01 oval	签到题 / 期望 微积分	≈ 900 / 872
12 pinball	前期题 / 模拟 几何	≈ 500 / 681
09 werewolf	前期题 / 推理 基环树	≈ 300 / 313
03 ringland	中期题 / 分析 枚举	≈ 250 / 30
04 shoot	中期题 / 区间DP	≈ 200 / 12
11 SACUL	中期题 / lucas 图论	≈ 100 / 30
02 bookshelf	中期题 / 容斥 gcd	≈ 80 / 114
10 chopping	中后题 / 枚举 双向链表	≈ 40 / 4
05 black	后期题 / 分析 基础DP	< 15 / 2
08 rectangle	后期题 / 扫描线 线段树	< 10 / 3
06 form	送命题 / ???	< 3 / 11
07 variance	送命题 / LCT	< 3 / 25

Tutorial

1001 oval and rectangle

把积分式子写出来 $\int_{y=0}^b (4 * y + 4a * \sqrt{1 - \frac{y^2}{b^2}}) dy$

换元, 设 $y = b \sin(\alpha)$, $dy = b \cos(\alpha) d\alpha$.

原式 $= \int_{\alpha=0}^{1/2\pi} (4b * \sin(\alpha) + 4a * \cos(\alpha)) b \cos(\alpha) d\alpha$

$= \int_{\alpha=0}^{1/2\pi} (2b^2 * \sin(2\alpha) + 2ab * (\cos(2\alpha) + 1)) d\alpha$

$= [-b^2 \cos(2\alpha) + ab(\sin(2\alpha) + 2\alpha)]_0^{1/2\pi}$

$= [b^2 + \pi ab] - [-b^2]$

$= 2b^2 - \pi ab$

最后除以 b .

$= 2b - \pi a$

1002 bookshelf

有 N 本一模一样的书，有一个共有 K 层的书架，现在要把书都放到书架上。

放完后假设第 i 层书架有 B_i 本书，则该层书架的稳固值为 $2^{B_i} - 1$ 。

定义整个书架的美观值为所有层书架的稳固值的 GCD 。

问现在随机放这些书，整个书架的美观值的期望值是多少。

Part 1

一个放 x 本书的层美观函数为 $b(x) = 2^{F_x} - 1$

整个书架的美观值为各层美观值的最大公约数，考察任意两项：

1. $gcd(b(x), b(y))$
2. $= gcd(2^{F_x} - 1, 2^{F_y} - 1)$
3. $= 2^{gcd(F_x, F_y)} - 1$
4. $= 2^{F_{gcd(x, y)}} - 1$

所以，对于 $n = x_1 + x_2 + \dots + x_k (x_i \geq 0)$ 的任一有序拆分方法来说：

1. 设他们的 gcd 为 $g = gcd(x_1, x_2, x_3 \dots x_k)$ 。
2. 书架的美观值即 $= 2^{F_g} - 1$
3. 所以我们枚举 g 然后只要能计算公共GCD是 g 的方案数有多少种即可。

Part 2

做法一：莫比乌斯反演

由数学期望的定义和基本的组合数学知识可得：

$$ans = \left(\sum_{t|n} (2^{F[t]} - 1) \sum_{t|p \& p|n} C_{n/p+k-1}^{n/p} \mu[p/t] \right) / C_{n+k-1}^n$$

做法二：直接对约数容斥

那么首先这个 g 一定是 n 的约数，所以我们直接对 n 的约数容斥就好了。

我们可以计算约数至少为 g 的方案数：

1. 每一层都是 g 的倍数，所以我们只要分配系数即可。
2. 系数的和为 $\frac{n}{g}$ ，共 k 个系数，每个非负，方案数是个经典组合数 $C(\frac{n}{g} + k - 1, k - 1)$ 。

然后容斥下把是 g 的倍数但不是 g , 的那些方案数扣掉。

就得到了严格是 g 的方案数, 所以从大的约数往小的约数做。

Part 3

最后就只剩下一个细节: 计算 2^{F_g} 这个东西。

由于 F_g 非常的爆炸, 所以要采用欧拉降幂。

1003 ring land

传说有一个世界, 这个世界由一个周长为 L 的环构成, 环上有 N 个新郎和 N 个新娘, 坐标已知。

现在要一一配对, 问所有配对方案中 N 对夫妻环上距离和 的最小值。

Part 0

是的, 这是 Codeforces 某道题的题面照搬 (理直气壮.jpg

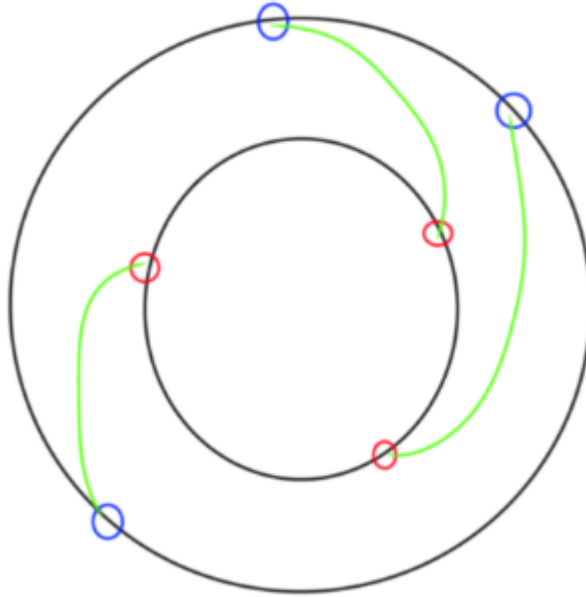
<http://codeforces.com/contest/981/problem/E>

原题是要求 (匹配方案中 N 对环上距离的最大值) 的最小值, 有兴趣可以去做做。

(出题人现场光荣的读成了现在这道题然后可劲码然后就没有然后了)

UPD: STD 做法涉及分析比较多, 事实证明各路神仙的写法更为优秀 (扑通

Part 1



首先可以看成靠得很近得两个环。

对于匹配在一起的点对，我们就把他们的最短路径画出来（逆时针或顺时针）。

然后容易发现：

1. 如果某两条路径出现了交叉（例图是不交叉的，如果压在一个环上可能会误以为交叉了）。
2. 那么让他们交换一下配偶（雾）。
3. 总的距离和不可能会变大，只可能会不变或变小。

所以如果某一对定下来了：

1. 比如第一个新郎和第三个新娘匹配了。
2. 那么第二个。。。四。。。就要匹配。
3. 依次推过去。
4. 这样总共其实只要考虑 N 个偏移量下答案各自是多少即可。

Part 2

然后就变成了一个可劲码出来就可以了的题。

比较直观的是点对的距离的贡献式子为以下其中一种：

1. $(a - b)$ 。
2. $(b - a)$ 。
3. $(a - b + L)$ 。
4. $(b - a + L)$ 。

具体是哪个主要受到

1. 坐标原点
2. b 落在 a 所在直径的哪一侧。

然后我们想象一下从小到大枚举偏移量的过程会发现

1. 一个点所在的点对的贡献式子只会最多把上面四个都遍历一次。
2. 把那些关键点抠出来然后暴力模拟过去维护就行。
3. 关键点的数量为 $O(n)$ 级别。

Part 3

当然这样可能还是不太好写（但总是能写的），还可以观察到：

1. 定义环上和一个点距离为 $L/2$ 的点为其对轴点。
2. 枚举偏移量的过程中，一旦某一个点的匹配点（和上一次的匹配点相比）越过了该点的对轴点。
3. 这个时候必然后出现路径交叉，除非当前或者不断的往大枚举偏移量后使得所有点的匹配都完成跨越对轴点。
4. 路径交叉一定可以调整，所以无视。
5. 所以我们可以通过这个观察简化。

即我们把 B 三倍空间掉（然后坐标 $+L$ 或 $-L$ ），所以贡献式子就只剩下：

1. $a - b$
2. $b - a$

相当于拿着 A 数组在三倍长的 B 数组上每挪一个位置比较一次。

对于每个点 a ，这个点只考虑匹配的点 b 落在区间 $[a - \frac{L}{2}, a + \frac{L}{2})$ 。

然后这些区间可以转成偏移量版本，最终得到 n 个偏移量区间，第 i 个是 $[l_i - i, l_i + n - i)$ 。

由上述的分析可知我们只要考虑偏移量为 $[\max_{i=1..n}(l_i - i), \min_{i=1..n}(l_i + n - i))$ 。

现在 a 的贡献就很好算了，把每个 a 对应的 b 之间大小关系发生反转的位置弄出来即可。

注意下枚举偏移量的时候， b 匹配的 a 点的改变方向是和 a 匹配的 b 点的改变方向相反的。

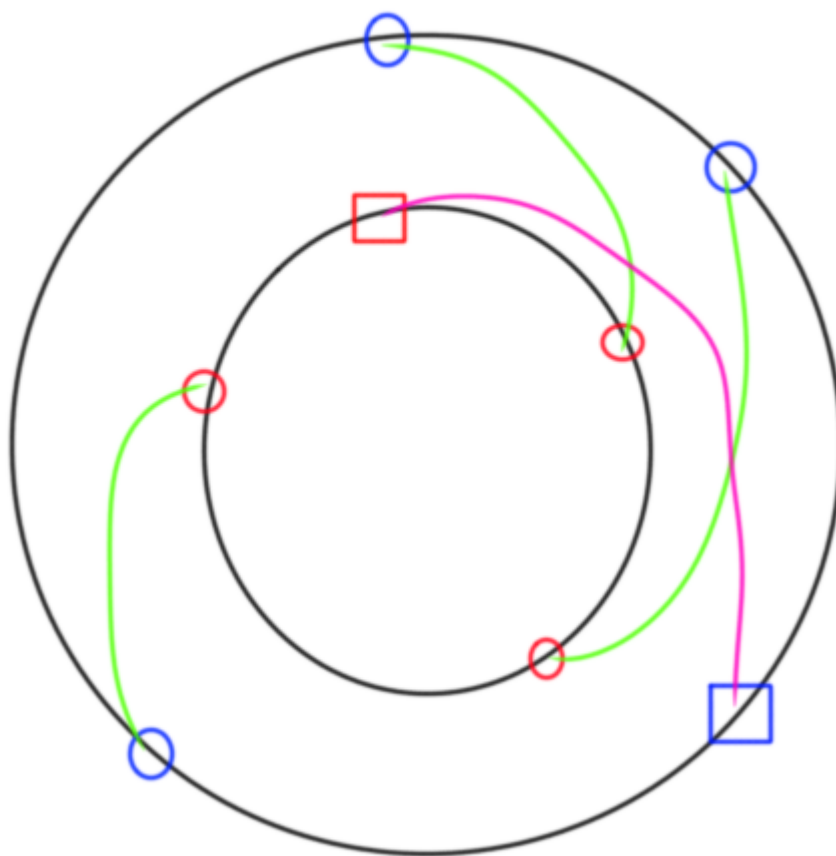
然后我们再努力感受下 b 的贡献就好。

详见 STD。

Part 4

如果对出现路径交叉总是可以在不增加距离和的前提下调整成不交叉（显然也是某个偏移量）有疑问：

1. 首先如果 N 只有 2，即两对，那把他们的匹配点交换即可。
2. 然后可以归纳一下，如果 $N - 1$ 对匹配的时候可以，那么：
 - 现在加一个点对（下图中的基佬紫路径），和 M 条路径发生了交叉。
 - 首先这 M 条路径彼此不交叉。
 - 依次交换即让这 M 条路径集体偏移即可（注意此时可能会出现跨越对轴点，但是先无视，即放任路径长度大于 $\frac{L}{2}$ ）。
 - 此时必然满足不交叉而距离和没有变大。



3. 然后我们每次抓一个跨越对轴点的路径（即路径长度超过 $\frac{L}{2}$ ）出来：
 - 把该路径反向，该路径和部分路径可能出现交。
 - 同第二步即可。
 - 第三步一定是有限次的过程，因为每次进行第三步都会导致距离和变小，而最终答案显然不可能无限小。
4. 推理完毕。

1004 Shoot Game

本题是一道区间动态规划的题目：

1. 只需要考虑像所有障碍物的两个端点去发射射线，就能包含所有的情况，那么一共只有 $2n$ 种射线。
2. 防御力最大的障碍物最终一定要被射穿，所以可以考虑每次从防御力最大的障碍物开始消灭。发射的射线会把沿途的其他障碍物一起消灭。

基于第一点，将所有端点按极角序进行离散化，端点下标为1到 $2n$ 。

设立状态 $dp(L, R)$, 表示消灭端点在 (L, R) 区间的所有线段, 所需要花费的最小能量。

基于第二点, 每次将区间内所包含的防御力最大的障碍物 v 找出来, 进行区间DP的转移:

$$dp(L, R) = \min\{dp(L, k-1) + dp(k+1, R) + W_v\}, (l_v \leq v \leq r_v)$$

最终的 $dp(1, 2n)$ 即是答案。

1005 black and white

$N * N$ 的格子棋盘, 每个格子为黑色或白色。

定义 special pair: 两个无法互达的黑色格子。

定义无法互达: 找不到一条包含起点终点, 仅由黑色格子构成, 且最多转向一次的路径。

一个 special pair 的得分为两个格子的曼哈顿距离。

问得分最大是多少, 以及有多少个 special pair 达到了这个得分。

Part 1

有一些 $N^2 \log^2 n$ 、 $\frac{N^3}{64}$ 的做法, 测试超时, 主要是大部分做法貌似都避不开要把图倒来倒去所以常数很大。

以及一些基于联通块的做法应该是不可行的(黑色格子稠密图)。

这题主要是要观察到一些性质, 分类讨论后就是一个基础DP。

不难察觉距离最大的黑色格子对很可能两个格子或至少其中一个格子在很边缘上。

Part 2

定义“行边缘点”: 当且仅当一个格子为其所在行的第一个或最后一个黑色格子。

定义“列边缘点”: 当且仅当一个格子为其所在列的第一个或最后一个黑色格子。

定义“边缘点”: 当且仅当一个点是“行边缘点”或“列边缘点”。

基于我们的猜想我们假设如果某 special pair 都是非边缘点。

```
D
.
C..A.....
.      .
.....B....E
      .
      .
      F
```


(A, B) 为我们所观察的 special pair。

$CDEF$ 为其他的黑色格子。

则如果该 pair 得分最大：

- AF 必然通。
- EA 必然通。
- CB 必然通。
- DB 必然通。

然后上述 4 个会推出 AB 必然通。

所以结论一：得分最大的 special pair 至少有一个点为“边缘点”。

Part 2

而可进一步观察：

- AB 是得分最大的 special pair。
- 如果 B 点是非边缘点。
- A 将具有哪些特性：

```
A$$$$$$$$$
#      .    $
#.....B....E
#      .
#      .
#####F
```

同样有：

- AE 相通，则 "\$" 路径必须通。
- AF 相通，则 "#" 路径毕竟通。

所以：

- B 左侧的 "." 路径至少必须有一个白格子，设为 (X_1, Y_1) 。
- B 上侧的 "." 路径至少必须有一个白格子，设为 (X_2, Y_2) 。
- 则 A 点必须在以 (X_2, Y_1) 为右下角的左上矩形内。
- 而反过来，在以 (X_2, Y_1) 为右下角的左上矩形内的黑色格子，也必然和 B 构成 special pair。

Part 3

然后其实就做完了，分类如下：

- 两个点都是边缘点：
 - 边缘点数量显然只有不到 $4N$ ，直接枚举判定即可。
- 仅有一个点是边缘点：

- 根据 Part2 的性质直接 $DP[i][j]$ 处理第 i, j 为右下角的左上矩形 $X + Y$ 最小的黑格子及数量。
- 处理每个格子往左、往上遇到的第一个白格子。
- 然后拿所有非边缘点出来查询一下即可。

沿 X/Y 轴翻转，做 4 次。

把同行同列的情况处理掉，然后这样讨论不会出现算重复的情况，所以不需要考虑去重的问题。

复杂度 $O(N^2)$ 级别，但是常数巨大。

1006 foam transformation

Part 1

先来分析一下操作 T 的含义是什么：

1. 假设 a, b, c 是三个等长数组，且 $a + b = c$ ，这里可看成向量。
2. 定义 a 的向前差分数组 $da[i] = a[i] - a[i - 1] (i \geq 2), da[1] = a[1]$
3. 那么必然有 $da + db = dc$ ，定义带入即可证明

操作 T 本质上是：选取数组的某个后缀 增加 or 减去有限平方序列，

因为平方函数是二阶等差数列，所以我们期望使用差分来想办法将这个增量数组化简为常量或者足够简单的形式。

我们考虑平方序列的差分

数组	1	2	3	4	5	6	7
a	1	4	9	16	25	36	49
da	1	3	5	7	9	11	13
dda	1	2	2	2	2	2	2
ddda	1	1	0	0	0	0	0

我们简写三阶差分为符号 D , $Da + Db = Dc$ 。

事实上：

- 将被操作的数组三阶差分之后。
- 每个操作 T 都可以看成是选择两个相邻的元素 $+1$ 或者同时 -1 。

这里我们要考虑一个陷阱：

- $T i k$, 中选择的是 $i, i + 1$ 。
- 因此 i 不能等于被操作序列的最大下标，即 $i < \text{序列长度 } m$ 。

这里题目描述中，已经要求如此

C 数组是目标数组，这里是全是 0 构成的等长数组

Part 2

我们对某个数组进行有限次相邻两个元素，同增同减的操作，

很容易发现数组的交错和守恒(奇数项和减去偶数项和)。

下面我们证明，这个条件是该数组可按照上述方法归零的充分必要条件

必要性

1. 从 0 出发开始构造数组。
2. 注意到每次操作之后，交错和不变，而刚开始0序列交错和为0。
3. 由数学归纳法可知只能构造出交错和为0的数组。
4. 必要性得证。

充分性：

1. 我们考虑构造法，假设一个数组满足交错和为 0。
2. 我们构造一种方法，使得该数组一定可以归零：
 1. 如果数组长度等于 1，仔细读题发现不存在这种情况。
 2. 如果数组长度等于 2，交错和为 0，则 $a[1] = a[2]$ ，显然直接可以达到目的。
 3. 如果数组长度大于 2，我们用一下方法，将下标大于2的元素值，一个单位一个单位的搬运到前两个元素。
 - 对 $i, i+1$ 同时 $+1$ ，然后 $i+1, i+2$ 同时 -1 ，这就使得 $i+2$ 位置的元素可以适配到 i 位置
 - 同理，同奇偶性的下标间，互相搬运是可以的。
 - 因此，奇数项我们可以搬到1，偶数项搬到2。
 - 由交错和为 0 知此时 $a[1] = a[2]$ ，回到第二种情况

Part 3

我们证明：

1. 一个交错和为0的数组如果以三个 0 结尾，也就是 $a[m] = a[m-1] = a[m-2] = 0$ 。
2. 那么三次差分以内，此数组交错和不变。

很明显的，如果两个等长数组相加，交错和为原来各自交错和之和。

我们设原来数组长度为 m ，是下面第一行

第一行	a_1	a_2	a_3	\dots	a_{m-2}	a_{m-1}	a_m	0
第二行	0	a_1	a_2	\dots	a_{m-3}	a_{m-2}	a_{m-1}	a_m

假设第一行交错和为 x 。

显然第二行交错和为 $-x$ 。

根据上面的结论，第一行减第二行得到的数组交错和为 $2x$ 。

但是注意到，事实上最后一列已经越界了，超出了 m 列的范围。

所以在真实的差分数组中，是不存在这一列的

所以我们多考虑了一个元素，真实的差分交错和是 $2x + a_m * (-1)^{(m+1)}$ 。

如果我们令 $a_m = 0$ ，那么结果就是 $2x$ 。

注意到，在差分过程中，数组最后一个元素的变化为 $a_m, a_m - a_{m-1}, a_m - 2a_{m-1} + a_{m-2}$ 。

如果我们令数组最后三个元素均为 0，就可以实现，三阶差分数组交错和为 $8x$ 。

命题得证

换句话说，只需要原来读入的数组区间交错和为 0，就是酸的，不再需要差分。

Part 4

为了方便起见，我们用个小技巧：

1. 将原来数组的偶数项全部乘上 -1 。
2. 这样原来数组的区间交错和，就是现在的区间和。

因此一个数组的酸度，就是这个新的数组有多少个区间和为 0。

考虑将这个变换后的数组求前缀和 s_i ($0 \leq i \leq n$)。

我们只需要算出在 $s[]$ 中，值 x 出现的次数 cnt_x (注意包括 $s_0 = 0$) 就可以知道数组酸度：

$$\text{酸度} = \sum_{x \in S} C_{cnt[x]}^2$$

这个用 `map` 很快就可以实现，因此直接就可以输出第一个答案。

下面考虑修改操作 $U\ i\ x$ ，含义是下标 $i, i+1$ 同时增加 x 。

很显然的，这个操作，相当于对我们变换后的数组， $a_i = a_i - x$ ， $a_{i+1} = a_{i+1} + x$ ，或者符号相反的操作。

稍加思考一下我们的 $s[]$ 数组，发现只有一个位置的数发生变化，那就是 a_i 。

只需要在 `map` 上修改一下两个点就可以了，维护一下答案输出即可。

1007 Variance-MST

暴力的思路是枚举平均值 A ，按照 $|w_e - A|$ 排序做最小生成树。

将边按照 w_e 排序，边(下标 i)会在一段连续的 A 上有贡献，右端点的值为 $\frac{w_i + w_j}{2}$ ，其中 $j(> i)$ 为最小的值满足图 e_i, \dots, e_j 中 e_i 位于一个环中，左端同理，使用LCT优化。

时间复杂度 $O(m \lg m)$ 。

1008 Rectangle Outline

考虑从下到上的扫描线。

假设当前遍历到一条矩形的出边：

1. 去掉它会导致一些区间原本有矩形覆盖，现在没有。
2. 这些区间肯定在矩形的轮廓上。
3. 可以用线段树把这些区间抠出来。
4. 四个方向都做一次扫描线，抠出所有轮廓。

可以证明轮廓线的拐点是不多于 $8 \cdot n$ 个的，证明如下：

1. 对于轮廓线上任意一个点：
 - 它要么在凸出去的一个角上，要么在凹进来的一个角上。
 - 凸出去的角只能由矩形的四个角贡献，所以最多 $4 \cdot n$ 个。
 - 凹进来的角个数 = 凸出去的角个数 - 4。
 2. 考虑按照题面描述的方向画轮廓线：
 - 若在直线行进方向右转，会生成一个凸出去的角。
 - 左转则生成一个凹进来的角。
 3. 最简情况下需要右转三次才能生成一条闭合折线。
 4. 每次左转都需要一次右转来抵消。
-

1009 Werewolf

N 个人，每个人可能好人可能坏人，每个人说了一句话，好人一定说了真话，坏人不一定。

每句话指定某个人是好人还是坏人，问哪些人一定是好人哪些人一定是坏人。

Part 1

首先，“所有人都是狼人”是合法的，所以第一个答案一定是0。

那么我们的任务就是确认有多少人满足存在一种方案使得这个人是村民，其他人就是铁狼了（即为第二个要求输出的答案）

我们把所有人抽象成点：

1. 对于每句话，建立一条说话的人指向被说的人的一条有向边。
2. 如果该句话为说某人是狼人，就称这条边为狼人边。

暂时不考虑狼人边，把图分成若干联通块，这样每个联通块：

1. 要么点数和边数一致（基环树）
2. 要么点数比边数多1（树）

对于基环树，由于没有狼人边，所以让他们都是村民是合法的，这些人都不是铁狼。

现在对于树，考虑狼人边（有且仅有一条），狼人边是树根连出去的。

1. 如果狼人边指向的是其他联通块：
 - 那么让树中的人都是村民，其他人都是狼人显然也合法。
 - 这些人都不铁狼。
2. 如果指向树中的某个节点：
 - 如果根是狼人，则其儿子就是狼人，以至于整棵树都是狼人，没有意义。
 - 如果根是村民，则其指向的节点 x 是铁狼，则以 x 为根的子树都是铁狼。
 - 而让树中的其它节点为村民，此时合法。
 - 所以树中非 x 子树的节点都可以是村民。

因此使用记忆化搜索把那些铁狼找出来，统计一下人数就行了。

时间复杂度 $O(n)$ 。

1010 Chopping hands

派大星想收藏一些实体卡牌。

牌店里总共还剩下 N 个卡包：

- 第 i 个卡包的价格为 C_i 。
- 其中有 D_i 张卡。
- 作为一个良心游戏，每个卡包里的卡片是哪些已知。

- 备注：每个卡包中可能含有重复的卡牌，且一个卡包被买走了就没了。

每当派大星获得一张卡牌，假设是第 i 种卡牌：

- 如果这种牌是第 j 次获得，则派大星觉得这张卡的价值就是 $Value_{i,j}$ 。

当派大星剁手完毕后：

- 先将派大星有的所有卡按照其价值排序，然后取其中位数 mid 。
- 派大星认为这些牌带给它的满意度 $V = 2 * mid$ 。
- 备注：中位数奇数个取正中间，偶数则取中间两个的平均。

设派大星共花了 S 元钱购买卡包，最终满意度为 V ，则派大星认为性价比为 $V - S$ 。

给定每个卡包的信息（价格 C_i ，卡数 D_i ，卡牌编号 $Card_{i,j}$ ($1 \leq j \leq D_i$)）。

给定每种卡牌重复拥有时的价值数组 $Value$ 。

派大星想知道性价比最大能达到多少。

Part 1

首先这个 n 的范围 2^1 赤裸裸的暗示了什么 2^N 的算法，事实也确实是如此。

由于每个卡包只能购买一次，所以总共方案也就是 2^N 。

然后我们使劲压这个暴力的复杂度就好了，可劲YY不同的状态间能共用那些内容。

Part 2

$O(2^{N-1} \times 100 \times \log 100)$ ：

1. 不难想到的是我们想让相邻两个状态最多就差一个 bit。
2. 这样只要维护这个 bit 带来影响，比如采用 **格雷码的遍历** 方式。
3. 然后如果裸跑，实际复杂度如上，中位数动态维护的方案有很多：
 - 两个优先队列倒一倒。
 - 线段树。
 - ...
4. 应该是会超时，100指的是卡牌数量， $\log 100$ 是维护中位数的复杂度

$O(2^N \times \frac{100}{N} \times \log 100)$ ：

复杂度分析是这题主要的一个设计点。

1. 翻转一个 bit 的代价是这包卡有多少张牌。
2. 观察格雷码的递归生成方法（当然也可以打代码跑计数

```
vector<int> gray(int n) {
    vector<int> v = {0, 1};
    for (int i = 1; i < n; i++)
        for (int j = v.size() - 1; j >= 0; j--)
            v.push_back(v[j] ^ (1 << i));
    return v;
}
```

3. 可知第 i 个 bit 被反转了 2^i 次, 所以让 100 张牌都在第 $n - 1$ 包会让上面一个做法超时。

进行了复杂度分析后不难想到我们把卡包按卡牌数量 (翻转代价) 重新排序一下:

- 此时对这种排序做法最坏的情况是所有卡包的数量平均。
- 所以此时反转的代价被均摊了, 为 $\frac{100}{N}$ 。
- N 是 20 所以大概比上一种最坏情况下提速 10 倍。

这种做法常数如果够小可能卡得过去, 因为 $\log 100$ 很小。

Part 3

$$O(2^N \times \frac{100}{N})$$

我们再努力努力把 $\log 100$ 扣掉。

观察中位数维护:

1. 中位数正常动态维护需要一个 \log , 但是有特例。
2. 如果只需要支持删除操作, 我们用双向链表维护中位数是 $O(1)$ 的。

不难联想到一颗基于 LOWBIT 构成的树:

1. 根节点是 2^N 。
2. 每个节点 i 的父节点即为 $i \text{ xor } \text{lowbit}(i)$ 。
3. 第 $i (0 \leq i < N)$ 位是 1 代表第 i 包不买, 0 代表买。 (根节点即为全部都买)
4. 往下走: 某一位 0 变 1, “买” 变成 “不买”, 本质是一个 **删除** 操作。
5. 往上走: 某一位 1 变 0, “不买” 变成 “买”, 本质是一个 **添加** 操作。

但是如果我们沿着这颗树 DFS 遍历一遍

1. 某一个时刻之所以可以往上走, 是因为之前往下走到了现在的位置。
2. 所以往上走的添加操作在这里可以改为 **撤销** 操作。
3. 所以我们只要记录 **删除操作** 修改了什么, 然后撤销掉就好了。
4. 类似一个栈, 往下走的时候一个删除操作进栈, 往上走的时候一个删除操作出栈。

至此维护 $O(1)$ 。

LOWBIT 树上第 i 位被反转了 2^{N-1-i} 次 (和格雷码刚好反过来)。

所以同样要进行 sort 来均摊成 $\frac{100}{N}$ 。

事实上撤销的复杂度常数甚至比删除来得小:

1. 写两句话把断掉的链表结点接回去。
2. 中位数直接记录然后 pop, 详见标程。

后记

推荐一下STD的DFS姿势。

以及虽然分析是这么分析，数据也做了加强，但是预测还是会有不少水过去的情况。

UPD：似乎没有水过去的情况，屎尿未及的拦住了学军AK。。

1011 sacul

应该不少人发现了把标题反过来读就是题解？（猜测脸.jpg

给定 i ，设 p 为第 i 个质数。

矩阵 A_n 为大小为 $p^n * p^n$ ，元素 $A_n[i][j] = (C[i][j] \% p) > 0$ 。

设 $F(n, k)$ 为矩阵 $(A_n)^k$ 的所有元素之和。

给定 n, k ，求 $\sum_{1 \leq i \leq n} \sum_{1 \leq j \leq k} F(i, j)$ ，对 $10^9 + 7$ 取模。

part 1

根据 Lucas 定理： $C(n, m) \% p = C(\frac{n}{p}, \frac{m}{p}) * C(n \% p, m \% p) \% p$ 。

其实本质上就是把 n, m 按进制 p 拆位——组合数后乘起来。

设其中一块是 $C(a, b)$ ($0 \leq a, b < p$)：

1. $a \geq b$ ，该组合数模 p 一定是正数（因为 a, b 都小于 p ）。
2. $a < b$ ，该组合数本来就是 0。

part 2

即 $C(n, m) \% p > 0$ 等价于 n, m 在 p 进制表示下， n 的每一位都不小于 m 的对应位。

而海绵宝宝 (HMBB) 是一个 01 矩阵：

1. 所以我们可以把它对应成一张有向图。
2. 更进一步可以理解成 $HMBB[i][j]$ 为从 i 出发，走 1 步，最终在 j 的方案数。
3. 海绵宝宝的 K 次方即对应成了走 K 步的方案数。

这张图的性质：

1. 根据最上面 Lucas 的分析，即 i 若在 p 进制下每一位都不小于 j ，则有一条 $i \rightarrow j$ 的边。

part 3

所以变成了一个组合数计数问题：

1. 现在要考虑 j 个共有 i 位的 p 进制数，且每个数的每一位不小于上一个数的对应位。
2. 显然 i 个位之间独立，所以 $F[i][j] = (C_{j+p}^{p-1})^i$ 。

所以 $ans = \sum_{j=1}^k \sum_{i=1}^n (C_{j+p}^{p-1})^i$

后面那部分是个等比数列求和。

时间复杂度 $O(K \log N)$ 。

有公比是 1 的情况，需要特别注意。

1012 Pinball

斜坡，给定小球初始位置，给定斜坡角度，长度给定重力加速度g，问小球在坡上弹了几次。

Part 1

小球落到斜坡上，垂直于斜坡的分速度会反向，推公式模拟这个过程即可。

可以二分也可以推公式求出下一次落点。

Part 2

针对不会物理选手，给一个计算几何方法

将重力加速度分解成垂直斜坡和平行斜坡两个方向。考虑这两个方向的分运动。

垂直斜坡方向，小球周期性的弹起下落。

平行斜坡方向，小球做匀加速直线运动。

只要算出垂直斜坡方向的分运动的周期，和平行斜坡方向分运动的总时间，除一下就是答案。

End.