题解

社团活动

数据点 1

显然，集会地点一定可以是某个活动人员的住址。

故对于每次活动，枚举活动地址，再枚举每个人，找到最小的距离和，输出即可。

复杂度： $O(n^2m)$

数据点 $2\sim 3$

从上个结论继续拓展，发现集会地点为所有人住址的中位数。（证明留作习题

故对于每次活动，枚举活动人员，找到中位数，即可找到最少的距离和，输出即可。

复杂度： $O(nm\log n)$

数据点 $4\sim 5$

（上一个做法卡卡常就过了，甚至不卡也过了）

沿用上一个数据点的做法，发现实际上我们只需要知道每个询问的中位数。

此后只需要求区间大于中位数的和以及小于中位数的和即可（注意自己处理偶数情况）

对此有许多 $O(nm+n^n)$ 的做法，可以自己思考。

数据点 $6\sim 7$

沿用上个数据点的思考，由于值域小，可以枚举值域，检查是否为中位数。

复杂度： $O(na+ma)$（此处 a 表示 a_i 的最大值）

数据点 $8\sim 10$

可以使用莫队处理中位数。有一个处理中位数的方法：用两个对顶堆，分别记录小于中位数的部分和大于中位数的部分，每次加入数的时候再判断两堆大小问题，做 $O(1)$ 次堆操作即可。

由于同时带增删的莫队比较好写，也可以考虑用平衡树代替堆或者堆上打tag。

可以发现这么做可以同时处理和，会比较好写。

复杂度： $O(n\sqrt{m}\log n)$

至此，本题就做完了，是一道莫队的模板题

中位数也可以用可持久化线段树求，时限可控制在$O(n\log n)$。

电梯

数据点 1

引理A：当物品重量均为 1 时，每次运输选择最大的 2^m 个即为最优

~~引理A证明留作习题~~

以引理A可以直接解决该数据点

数据点 $2\sim3$

由引理A可解决该数据点， c_i 较大时使用除法、取模等即可。

数据点 $4\sim8$

引理B：一个重量为 2 的货物等价于两个重量为 1 的货物

引理B证明如下：

先假设重量 2 的货物可以拆分为两个重量为 1 的货物，以此安排方案。

此时可能存在一个重量 2 的货物可能被安排在两次运输。可以考虑将该货物完整安排在后一次运输，而将后续的第一个重量 1 的货物移至该次运输。经过该操作，可以发现每次运输均未超载，且耗时不变。

注：此结论依赖于 $n\%2=0$

经过引理B，可将该数据点转化为数据点 $2\sim3$ 的条件

数据点 $8\sim13$

引理C：一个重量为 2^m 的货物等价于 2^m 个重量为 1 的货物

证明类似引理B

数据点 $14\sim20$

用 $bitset$ 或 $bool$ 数组即可。

抢凳子

以下请自行注意取模、爆`int`等问题

数据点 1

没有加分操作，所以输出 n 个 0 即可。

数据点 $2\sim3$

没有乘法操作，所以人员的相对位置不变，只需要记录位移。

主要伪代码如下：

加法操作：`p+=x;`

加分操作：`ans[x-p]++;`

数据点 $4 \sim 5$

$O(nm)$ 大暴力，不讲了

数据点 $6 \sim 8$

先不考虑乘 n 的操作

由 费马小定理？ 得知不会出现抢凳子的情况

所以只需要记录目前总的乘法和加法数值，加分时倒推即可

主要伪代码如下：

加法操作：`p+=x;`

乘法操作：`t*=x;p*=x;`

加分操作：`ans[(x-p)*(t**(n-2))]++`

如果出现了乘 n 的操作，那乘 n 后只剩一个人未被淘汰，所以只需要暴力解决即可。

数据点 $9 \sim 11$

与 **数据点** $6 \sim 8$ 基本一致，区别在于求逆元需要用 $exgcd$

数据点 $12 \sim 25$

从 **数据点** $9 \sim 11$ 拓展，仅需考虑 x, n 不互质的情况。

（~~经过冥思苦想之后，发现学过的数学好像没讲过能快速处理谁抢到凳子这个问题~~）

（~~于是开始查数论、数学~~）

（~~然后发现还是没办法快速处理~~）

所以考虑当 x, n不互质 时暴力处理，此时的一次操作是 $O(n)$ 的，可以发现此次操作后至少淘汰 $n/2$ 的人。

可以发现未被淘汰的人的间隔是固定的，所以基本可以看作是一个 n/gcd 规模的游戏。

注意一下实现方法，复杂度就可以优化到 $n + \frac{n}{2} + \frac{n}{4} + \cdots + 1 = O(n)$ 。所以在这种情况下，暴力处理的复杂度是可接受的。

另：如果实现方式不好，复杂度会变成 $O(n \log n)$ ，仍然能过这题

高速收费

数据点 1

由于只有两列点，所以所有纵向的边都没有用。只需考虑横向的边。

故可以看作有 n 条互不影响的路径，故每次操作必然选择最短的一条。

故仅需找到最小值，算出所需长度即可。

数据点 2

本数据点没有优秀的做法，仅供各种朴素算法。

数据点 3

注意到只需在每一行的第一条横边操作一次就可以将最短路增加1，此时答案是 $n*k$ ，可以证明没有更优的做法。

数据点 4

看到数据范围，猜测是网络流？但是不会流？

对于这个数据点，将所有最短路的路径找出来，要求用最少的操作覆盖所有的路径。

其中一种想法就是用网络流跑最小割，也就解决了这个数据点。

数据点 5

与 **数据点** 4 基本一致，但代码难度直线上升（

数据点 $6 \sim 10$

可以映射到平面的图的最短路和最小割可以互相转化。在 OI 中，网格图的转化较为常见。例如 [P7916 CSP-S 2021 交通规划]。这个转化的正确性请自行搜索（P7916 的题解里就有）

看回我们这道题，先求出目标的最短路长度 K 。考虑对偶后的图，由最小割等于最大流，我们现在希望这个图的最大流达到 K ，我们的每次操作可以让一条边的流量 $+1$ ，每条边的操作次数没有限制。也就等价于给每条边增加一条流量 $\inf$ 单位流量花费 1 的“副边”。跑流量为 K 的最小费用即可解决本题。