

2024 NOIP模拟赛

by GDSY

题目名称	社团活动	电梯	抢凳子	高速收费
题目类型	传统型	传统型	传统型	传统型
目录	activity	lift	chair	charge
可执行文件名	activity	lift	chair	charge
输入文件名	activity.in	lift.in	chair.in	charge.in
输出文件名	activity.out	lift.out	chair.out	charge.out
每个测试点时限	2.0秒	1.0秒	1.0秒	2.0秒
内存限制	128M	512M	128M	128M
测试点数目	10	20	25	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于C++语言	activity.cpp	lift.cpp	chair.cpp	charge.cpp

编译选项

对于C++语言	-O2 -std=c++14 -static

注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
- 提交的程序代码文件的放置位置请参考各省的具体要求。
- 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
- 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
- 只提供 Linux 格式附加样例文件。
- 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

T1 社团活动 (activity)

时空限制

时间限制：2000 ms

空间限制：128 MB

题目背景

你参加了一个社团，要策划每个周末的社团活动，需要让社员们的路程和尽可能少。

这很难，所以你决定先考虑真空中的球形鸡社员住在数轴上的情况。

题目描述

你们社团共 n 个人，编号为 $1 \sim n$ 。第 i 个人住在数轴上 a_i 的位置。

现在要规划接下来 m 周的活动规划。第 i 周需要编号为 $[l_i, r_i]$ 的成员参加活动。

每次活动需要在数轴上选定一个点作为活动地点，使得所有参加活动的人从住处到活动地点的距离和最小，你需要输出这个距离和。

答案可能很大，请输出其对 2^{32} 取模的结果。

输入格式

第一行两个整数 n, m ，分别表示社团总人数和活动的次数。

第二行 n 个整数 a_i ，表示每个人住的地方。

接下来 m 行，每行两个整数 l_i, r_i ，表示第 i 次活动参加人员的编号区间。

输出格式

m 行，每行一个整数，表示第 i 次活动的距离和

样例组

见附件

数据范围

$$1 \leq n, m \leq 50000$$

$$1 \leq a_i \leq 50000$$

$$1 \leq l_i \leq r_i \leq n$$

具体每个测试点数据范围如下，留空表示无额外限制。

数据点编号	n	m	a_i	其他限制
1	≤ 400	≤ 400		
2 ~ 3	≤ 1000	≤ 1000		
4 ~ 5	≤ 3000	≤ 3000		
6 ~ 7			≤ 200	
8 ~ 10				

T2 电梯 (lift)

时空限制

时间限制：1000 ms

空间限制：512 MB

题目背景

出题人想题目背景想到快疯掉了，建议选手自己想。

题目描述

在大楼一楼有 n 组货物，现需用一电梯将货物运至对应楼层。

第 i 组货物有 c_i 件，每件货物重量为 2^{w_i} ，需要送至第 f_i 层。

电梯单次运输载重上限为 2^m 。

每次运输电梯从一楼出发，运输一些货物，上升路上在对应楼层卸货，直到所有货物卸下，再返回一楼，总耗时为这次运输电梯到达最高的楼层。

形式化的说，我们将一件货物视为一个二元组 (w, f) ，每次运输视为一个可重集 S ，要求 $\sum_{(w,f) \in S} w \leq 2^m$ ，耗时为 $\max_{(w,f) \in S} f$ 。

注：同一组货物并不要求同一次运输，可以视为共有 $\sum c_i$ 件互相独立的货物

请计算将所有货物运至各自目的地所需的最少时间。

输入格式

第一行两个整数 n, m ，分别表示货物组数和电梯运量。注：电梯运量为 2^m 。

接下来 n 行每行三个整数 c_i, w_i, f_i ，分别表示第 i 组货物的数量、单重与目的楼层。注：货物单重为 2^{w_i} 。

输出格式

一行一个整数，表示需要的最少时间。

样例组

见附件

数据范围

$$1 \leq n, m, c_i, f_i \leq 5 \times 10^5$$

$$0 \leq w_i \leq m$$

具体每个测试点数据范围如下，留空表示无额外限制。

数据点编号	n	m	c_i	w_i
1			$= 1$	$= 0$
2 ~ 3				$= 0$
4 ~ 5			$= 1$	≤ 1
6 ~ 7				≤ 1
8 ~ 10		≤ 30		
11 ~ 13		≤ 60		
14 ~ 20				

T4 抢凳子 (chair)

时空限制

时间限制：1000 ms

空间限制：128 MB

题目背景

你觉得原版抢凳子太简单，所以你决定给这个游戏多加点规则。

题目描述

有编号从 0 到 $n - 1$ 的 n 张凳子按编号顺序顺时针围成一圈。初始时有 n 个玩家，编号为 i 的人坐在 i 号凳子上（特别的，编号为 n 的玩家初始时坐在 0 号凳子上）。

接下来，裁判发出 m 次指令。指令分为以下三种：

1. 给在 $x \% n$ 号凳子的玩家加一分（若指定凳子没有玩家，则不加分）
2. 要求每个人按照顺时针方向移动 x 步，也就是说原本在 i 号凳子的玩家需要移动到 $(x + i) \% n$ 号凳子处。
3. 要求每个人移动到自己所在凳子编号 x 倍的位置，也就是说原本在 i 号凳子的玩家需要移动到 $(x * i) \% n$ 号凳子处。

特别的，如果某次指令要求多个玩家移动到同一张凳子，那么原本位置距离该凳子顺时针距离最短的玩家能抢到这凳子，而其他被要求移动到这张凳子的玩家被淘汰，无法再获得分数。（但已获得的分数不会清空）

但是这个游戏太复杂了，没人想玩，只能请你写一个程序来计算每个玩家最终的得分。

输入格式

第一行两个整数 n, m ，分别表示凳子数量和指令数量

接下来 m 行，每行两个整数 $type, x$ ，分别表示指令类型和指令内容

输出格式

一行 n 个整数，第 i 个整数表示 i 号玩家的最终得分

样例组

见附件

数据范围

$$1 \leq x \leq n \leq 5 \times 10^5$$

$$1 \leq q \leq 5 \times 10^5$$

$$1 \leq type \leq 3$$

具体范围如下（留空为无特殊限制）

数据点编号	n	m	特殊条件
1			$type \neq 1$
2 ~ 3			$type \neq 3$
4 ~ 5	≤ 3000	≤ 3000	
6 ~ 8			n 为质数
9 ~ 11			当 $type = 3$ 时, x 与 n 互质
12 ~ 25			

T4 高速收费 (charge)

时空限制

时间限制：2000 ms

空间限制：128 MB

题目背景

你管理着一个地区的高速公路，现在由于你们公司的财政危机，你希望增加高速费。但是为了避免舆论危机，你又不大幅提高费用。因此，你将他抽象成以下问题，让计算机来计算。

注：题目背景仅为胡编乱造，事实上高速收费多少有国家文件规定，请勿乱改。

题目描述

这片地区的每个地点可以视为在平面上的 $n * m$ 个格点，第 i 行 j 列的格点记为 (i, j) ，相邻（四联通）的格点之间有一条高速路段相连。

每段高速有个原定的收费，而一次行程的收费为路程上的高速路段的收费和。

为了增收，你可以进行若干次操作，每次可以使一个路段的收费 $+1$ 。你希望从第 1 列到第 m 列的最小花费比原来提高 k 。

由于舆论压力，你希望最小化操作次数。

请输出这个操作次数。

输入格式

第一行两个整数 n, m ，表示这个地区可以抽象为 n 行 m 列的格点。

接下来 n 行，每行 $m - 1$ 个整数，其中第 i 行的第 j 个整数表示 (i, j) 到 $(i, j + 1)$ 的收费。

接下来 $n - 1$ 行，每行 m 个整数，其中第 i 行的第 j 个整数表示 (i, j) 到 $(i + 1, j)$ 的收费。

最后一行 1 个整数 k ，表示你希望最小花费提升的数值。

输出格式

一行一个整数，表示最小的操作次数。

样例组

见附件

数据范围

$$2 \leq n, m \leq 500$$

$$2 \leq n * m \leq 1000$$

$$0 \leq a_{ij}, b_{ij} \leq 2000 \text{ (} a, b \text{ 表示初始收费)}$$

$$1 \leq k \leq 4000$$

具体数据范围如下（留空表示无特殊限制）：

数据点编号	n	m	$n * m$	k	a_{ij}, b_{ij}
1		$= 2$			
2			≤ 20		
3					$= 1$
4				$= 1$	
5				$= 2$	
6 ~ 10					