

CCF 全国青少年信息学奥林匹克联赛

NOIP 模拟赛 2025-11-22

时间：2025 年 11 月 22 日 18:30 ~ 21:30

题目名称	Longge 的信号塔	Longge 的双核服务器	Longge 的情报网	Longge 的量子矩阵
题目类型	传统型	传统型	传统型	传统型
目录	<code>tower</code>	<code>dualcore</code>	<code>network</code>	<code>matrix</code>
可执行文件名	<code>tower</code>	<code>dualcore</code>	<code>network</code>	<code>matrix</code>
输入文件名	<code>tower.in</code>	<code>dualcore.in</code>	<code>network.in</code>	<code>matrix.in</code>
输出文件名	<code>tower.out</code>	<code>dualcore.out</code>	<code>network.out</code>	<code>matrix.out</code>
每个测试点时限	1.0 秒	2.0 秒	1.0 秒	2.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	20	20	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言 | `tower.cpp` | `dualcore.cpp` | `network.cpp` | `matrix.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。
- `main` 函数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。
- 本次测评全部采用子任务捆绑测试。

A. Longge 的信号塔 (tower)

题目背景

Longge 刚刚入职了 Dachuli 通信公司，担任首席网络架构师。公司最近要在一条笔直的数轴公路上部署信号覆盖。公路上有无数个整数坐标点。

为了节省成本，Dachuli 并没有直接建设基站，而是选择向第三方供应商 XiaochuZhang 购买现成的“信号频段”。市场上每天都会推出一个新的频段供 Longge 挑选。

题目描述

XiaochuZhang 的商店里一共有 n 个信号频段。第 i 个频段覆盖了从 l_i 到 r_i 的所有整数坐标（包含端点），售价为 c_i 枚金币。

Longge 每天都会去商店看看前 s 个已上架的频段 ($1 \leq s \leq n$)。作为架构师，Longge 深知“信号共振”原理：

如果你购买了一组频段，且这组频段中包含的最小坐标为 L_{min} ，最大坐标为 R_{max} ，那么由于共振效应，所有位于闭区间 $[L_{min}, R_{max}]$ 内的整数坐标点都会被信号完美覆盖。

Longge 的目标是：对于每一个 s （即只考虑前 s 个频段时），他希望构建的信号网络能够覆盖**当前所有可用频段能达到的最大范围**。具体来说，如果前 s 个频段中所有 l_j 的最小值为 $MinL$ ，所有 r_j 的最大值为 $MaxR$ ，那么 Longge 购买的方案必须使得最终覆盖的范围至少包含 $[MinL, MaxR]$ 。

在满足上述覆盖范围的前提下，Longge 希望花费的金币最少。请你帮助 Longge 计算出， s 从 1 到 n 每一天所需的最小花费。

输入格式

第一行包含一个整数 t ($1 \leq t \leq 1000$)，表示测试数据组数。

对于每组测试数据：第一行包含一个整数 n ($1 \leq n \leq 10^5$)，表示频段的总数量。接下来的 n 行，每行包含三个整数 l_i, r_i, c_i ($1 \leq l_i \leq r_i \leq 10^9, 1 \leq c_i \leq 10^9$)，分别表示第 i 个频段的左端点、右端点和价格。

保证所有测试用例中 n 的总和不超过 $2 \cdot 10^5$ 。

输出格式

对于每组测试数据，输出一行 n 个整数。第 s 个整数表示：如果只提供前 s 个频段，Longge 达成最大覆盖范围所需的最小花费。

样例 #1

输入 #1

```
3
2
2 4 20
7 8 22
2
5 11 42
```

```
5 11 42
6
1 4 4
5 8 9
7 8 7
2 10 252
1 11 271
1 10 1
```

输出 #1

```
20
42
42
42
4
13
11
256
271
271
```

说明/提示

样例解释

在第一个测试用例中：

- 当 $s = 1$ 时，只有一个频段 $[2, 4]$ ，花费 20。最大范围即为 $[2, 4]$ 。
- 当 $s = 2$ 时，可用频段为 $[2, 4]$ (20金币) 和 $[7, 8]$ (22金币)。
 - 全局最小左端点 $MinL = 2$ 。
 - 全局最大右端点 $MaxR = 8$ 。
 - 为了覆盖 $[2, 8]$ ，Longge 必须购买这两个频段（因为没有单个频段能同时覆盖2和8），总花费 $20 + 22 = 42$ 。根据共振原理，他获得了 $[2, 8]$ 之间的所有点。

数据范围与约定

本题共设 5 个子任务，只有通过该子任务下的所有测试点才能获得对应分数。

子任务	分值	n 的限制	特殊性质 (Special Nature)
1	10	$\sum n \leq 50$	无
2	20	$\sum n \leq 2 \cdot 10^5$	保证对于所有 i ，都有 $l_i = 1$
3	20	$\sum n \leq 2 \cdot 10^5$	保证对于所有 i ，都有 $l_i = r_i$
4	20	$\sum n \leq 2000$	无
5	30	$\sum n \leq 2 \cdot 10^5$	无

B. Longge 的双核服务器 (dualcore)

题目背景

Longge 最近被任命为一家大型科技公司的云计算部门主管。为了应对日益增长的算力需求，Longge 引入了两套高性能计算集群，分别由公司的两位技术大牛 DachuLi 和 XiaochuZhang 负责维护。

DachuLi 负责的是代号为 "Fire" 的 CPU 密集型计算集群，每秒可以提供 f 单位的算力。

XiaochuZhang 负责的是代号为 "Water" 的 GPU 密集型计算集群，每秒可以提供 w 单位的算力。

虽然两个集群的架构不同，但它们经过了特殊的容器化封装，可以处理通用的计算任务。

题目描述

Longge 手头现在积压了 n 个待处理的计算任务。第 i 个任务需要的总运算量为 s_i 。

为了完成这些任务，Longge 需要将它们分配给 "Fire" 或 "Water" 集群。分配规则如下：

1. 每个任务必须完整地分配给其中一个集群，不可拆分。
2. 一旦任务被分配给某个集群，该集群需要消耗等同于任务运算量的算力来完成它。例如，如果任务 i 分配给 "Fire" 集群，则需要消耗 s_i 单位的 Fire 算力；若分配给 "Water" 集群，则消耗 s_i 单位的 Water 算力。

系统从第 0 秒开始运行。随着时间的推移，两个集群会不断积累可用的算力资源（可以理解为集群完成了之前的后台维护，释放了算力配额）。具体来说，在第 t 秒结束时，"Fire" 集群累计可支配的算力上限为 $t \times f$ ，"Water" 集群累计可支配的算力上限为 $t \times w$ 。

Longge 希望知道，最少需要经过多少秒，两个集群累计的算力足以处理完所有分配给它们的任务？换句话说，你需要找到最小的整数时间 T ，使得存在一种将 n 个任务划分为集合 S_{Fire} 和 S_{Water} 的方案，满足：
$$\sum_{j \in S_{Fire}} s_j \leq T \times f \quad \text{且} \quad \sum_{k \in S_{Water}} s_k \leq T \times w$$

注：Longge 是算法大师，对于给定的时间 T ，他总能瞬间判断出是否存在合法的分配方案。

输入格式

第一行包含一个整数 t ($1 \leq t \leq 100$)，表示测试数据组数。

对于每组测试数据：第一行包含两个整数 w, f ($1 \leq w, f \leq 10^9$)，分别表示 XiaochuZhang (Water) 和 DachuLi (Fire) 集群每秒产生的算力。第二行包含一个整数 n ($1 \leq n \leq 100$)，表示任务的数量。第三行包含 n 个整数 s_1, s_2, \dots, s_n ($1 \leq s_i \leq 10^4$)，表示每个任务所需的运算量。

保证所有测试用例中 n 的总和不超过 100。

输出格式

对于每组测试数据，输出一个整数，表示完成所有任务所需的最短时间（单位为秒）。

样例 #1

输入 #1

```
4
2 3
3
2 6 7
37 58
1
93
190 90
2
23 97
13 4
4
10 10 2 45
```

输出 #1

```
3
2
1
5
```

说明/提示

样例解释

在第一个测试用例中： $w = 2, f = 3$ 。任务大小为 2, 6, 7。

- 如果 $T = 1$ ，Water 算力 2，Fire 算力 3。无法处理任何任务组合。
- 如果 $T = 2$ ，Water 算力 4，Fire 算力 6。无法同时放下 6 和 7。
- 如果 $T = 3$ ，Water 总算力 $3 \times 2 = 6$ ，Fire 总算力 $3 \times 3 = 9$ 。
 - 方案：将大小为 6 的任务分配给 Water（消耗 6，剩余 0）；将大小为 2 和 7 的任务分配给 Fire（消耗 $2 + 7 = 9$ ，剩余 0）。
 - 满足条件，故最小时间为 3。

数据范围与约定

本题共设 5 个子任务，只有通过该子任务下的所有测试点才能获得对应分数。令 $S = \sum_{i=1}^n s_i$ 表示所有任务的总运算量。

子任务	分值	n	S	特殊性质
1	10	≤ 10	≤ 200	无
2	15	≤ 100	$\leq 10^6$	保证所有 $s_i = 1$
3	20	≤ 100	≤ 1000	无
4	20	≤ 100	$\leq 10^6$	保证 $w = f$
5	35	≤ 100	$\leq 10^6$	无

C. Longge 的情报网 (network)

题目背景

在 D 国的地下抵抗组织中，情报的安全传递至关重要。Longge 作为组织的情报总长，刚刚收到了一份绝密文件，需要尽快将其从代号为 1 的特工手中传递给代号为 M 的特工。

组织的通信网络由 XiaochuZhang 负责维护。为了防止被敌方监听，整个组织被划分成了 N 个独立的“情报小组”。每个小组由若干名特工组成。根据组织的安全协议：

1. 同一个小组内的特工可以直接共享情报。
2. 不同小组之间**严禁**直接通信。
3. 唯一一种打破隔阂的方式是“小组融合”：如果两个小组 X 和 Y 中至少拥有一名共同的特工（即有人同时拥有这两个小组的身份），那么 Longge 可以下令将这两个小组进行**融合**。
4. 融合后的新小组 $X \cup Y$ 包含原两个小组的所有成员，且所有成员之间均可互通情报。

为了减少组织架构变动带来的暴露风险，Longge 希望进行最少次数的“小组融合”操作，使得特工 1 和特工 M 最终能够处于同一个（融合后的）大组中，从而完成情报传递。

题目描述

XiaochuZhang 提供了一份目前所有小组的名单。一共有 N 个小组，第 i 个小组包含 A_i 名特工，特工的代号为 $S_{i,1}, S_{i,2}, \dots, S_{i,A_i}$ （代号在 1 到 M 之间）。

你可以进行任意次（包括零次）如下操作：

- 选择两个当前存在的小组 X 和 Y ，满足 $X \cap Y \neq \emptyset$ （即至少有一个公共成员）。
- 将 X 和 Y 从名单中移除，并将 $X \cup Y$ 加入名单。

请计算：最少需要进行多少次操作，才能使得最终名单中存在一个小组，同时包含特工 1 和特工 M ？如果无论如何都无法实现，请输出 -1 。

输入格式

第一行包含两个整数 N 和 M ，分别表示小组的数量和特工代号的最大值。

接下来 $2N$ 行，描述 N 个小组的信息：对于第 i 个小组 ($1 \leq i \leq N$)：

- 第一行包含一个整数 A_i ，表示该小组的特工人数。
- 第二行包含 A_i 个整数 $S_{i,1}, S_{i,2}, \dots, S_{i,A_i}$ ，表示该小组内特工的代号。

输出格式

输出一个整数。如果可以实现目标，输出最少的操作次数；否则输出 -1 。

样例 #1

输入 #1

```
3 5
2
1 2
2
2 3
3
3 4 5
```

输出 #1

```
2
```

样例 #2

输入 #2

```
1 2
2
1 2
```

输出 #2

```
0
```

样例 #3

输入 #3

```
3 5
2
1 3
2
2 4
3
2 4 5
```

输出 #3

```
-1
```

样例 #4

输入 #4

4 8
3
1 3 5
2
1 2
3
2 4 7
4
4 6 7 8

输出 #4

2

说明/提示

样例解释 1

- 初始小组: $S_1 = \{1, 2\}, S_2 = \{2, 3\}, S_3 = \{3, 4, 5\}$ 。
- 第一次操作: 选择 S_1 和 S_2 (公共成员 2), 融合得到 $\{1, 2, 3\}$ 。
- 第二次操作: 选择 $\{1, 2, 3\}$ 和 S_3 (公共成员 3), 融合得到 $\{1, 2, 3, 4, 5\}$ 。
- 此时 1 和 5 在同一组中。共需 2 次操作。

样例解释 2

- 初始小组 $S_1 = \{1, 2\}$ 已经同时包含 1 和 2。无需操作, 答案 0。

数据范围与约定

本题共设 4 个子任务, 只有通过该子任务下的所有测试点才能获得对应分数。

子任务	分值	N, M	$\sum A_i$	特殊性质 (Special Nature)
1	10	≤ 10	≤ 50	无
2	25	$\leq 2 \cdot 10^5$	$\leq 5 \cdot 10^5$	保证所有 $A_i = 2$
3	25	≤ 1000	≤ 2000	无
4	40	$\leq 2 \cdot 10^5$	$\leq 5 \cdot 10^5$	无

特殊性质说明:

- $A_i = 2$: 表示每个小组恰好只有两名特工。这种情况下, 每个小组可以看作是连接两名特工的一条无向边。

通用约束:

- $1 \leq N \leq 2 \times 10^5$
- $2 \leq M \leq 2 \times 10^5$
- $1 \leq \sum_{i=1}^N A_i \leq 5 \times 10^5$
- $1 \leq S_{i,j} \leq M$
- 每一组内的成员代号互不相同。

D. Longge 的量子矩阵 (matrix)

题目背景

Longge 的科技帝国最近涉足了量子计算领域。作为首席技术官，Longge 设计了一款新型的“双态量子处理器”，其核心是一个 N 行 M 列的量子矩阵。

为了维持矩阵的稳定性，必须在其中的网格点上部署两种特殊的量子稳定器：

- 由 DachuLi 研发的“黑曜石 (Black)”稳定器，共需部署 B 个。
- 由 XiaochuZhang 研发的“白象牙 (White)”稳定器，共需部署 W 个。

然而，这两位大牛的设备存在严重的兼容性问题。根据量子纠缠的“飞车原理” (Rook's Principle)：

- 如果一个“黑曜石”稳定器和一个“白象牙”稳定器处于同一行或同一列，它们之间会产生剧烈的能量干扰（互相攻击），导致整个矩阵崩溃。
- 同种颜色的稳定器之间由于频率同步，处于同行同列不会发生干扰。

Longge 现在面临的任务是：在一个 $N \times M$ 的空矩阵中，找出所有合法的部署方案，使得全部 B 个黑曜石稳定器和 W 个白象牙稳定器都被安放，且任意一个黑稳定器和任意一个白稳定器之间都不会产生干扰。

题目描述

给定一个 N 行 M 列的网格。你需要将 B 个黑棋子和 W 个白棋子放置在网格中。

一个合法的“好配置”必须满足以下所有条件：

- 所有的 $B + W$ 个棋子都必须放置在网格内。
- 每个格子上最多只能放置一个棋子。
- 互不侵犯**：不存在任何一对（黑棋子，白棋子），使得它们处于同一行或同一列。

请计算有多少种不同的“好配置”方案。由于答案可能非常大，请输出方案数对 998244353 取模的结果。注：同颜色的棋子之间不区分个体（即视为全同粒子），但黑色和白色棋子是不同的。

输入格式

输入包含一行，由四个空格分隔的整数组成： N M B W 分别表示矩阵的行数、列数，黑棋子数量，白棋子数量。

输出格式

输出一个整数，表示满足条件的方案数模 998244353 的结果。

样例 #1

输入 #1

```
2 2 1 1
```

输出 #1

4

样例 #2

输入 #2

1 2 1 1

输出 #2

0

样例 #3

输入 #3

40 40 30 30

输出 #3

467620384

说明/提示

样例解释 1

在 2×2 的网格中放 1 黑 1 白。为了不冲突，它们不能同行也不能同列。合法的放法是对角线放置：

- 黑(1,1), 白(2,2)
- 黑(2,2), 白(1,1)
- 黑(1,2), 白(2,1)
- 黑(2,1), 白(1,2) 共 4 种。

样例解释 2

在 1×2 的网格中，无论怎么放，1 黑 1 白必然处于同一行（第 1 行），因此一定会冲突。答案为 0。

数据范围与约定

本题共设 5 个子任务，只有通过该子任务下的所有测试点才能获得对应分数。

子任务	分值	N, M	B, W	特殊性质 (Special Nature)
1	10	≤ 4	≤ 16	无
2	10	≤ 50	$W = 0$	只需要放置黑棋子
3	10	≤ 50	$B = 1, W = 1$	只需放置 1 黑 1 白
4	30	≤ 20	≤ 400	无
5	40	≤ 50	≤ 2500	无

特殊性质说明：

- $W = 0$ ：此时不存在白棋子，因此条件 3（互不侵犯）自动满足。问题转化为在 $N \times M$ 的格子中选 B 个位置放黑棋子。
- $B = 1, W = 1$ ：问题转化为在 $N \times M$ 的格子中选两个位置，使得它们既不同行也不同列。

通用约束：

- $1 \leq N, M \leq 50$
- $1 \leq B, W \leq 2500$
- $B + W \leq N \times M$
- 输入均为整数