全国青少年信息学奥林匹克联赛

NOIP 2024

模拟赛 *II*

water_tomato

题目名称	货币代码	Sumplete 游戏	高速公路	加数
题目类型	传统型	传统型	传统型	传统型
目录	coin	game	road	add
可执行文件名	coin	game	road	add
输入文件名	coin.in	game.in	road.in	add.in
输出文件名	coin.out	game.out	road.out	add.out
每个测试点时限	1.0 秒	2.0 秒	2.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	10	10	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	coin.cpp	game.cpp	road.cpp	add.cpp

编译选项

对于 C++ 语言	-O2 -std=c++20 -DOFFLINE_JUDGE

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 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。
3. 因违反以上两点而出现的错误或问题，申诉时一律不予受理。
4. 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）。
5. 选手提交的程序源文件必须不大于 100KB。
6. 程序可使用的栈空间内存限制与题目的内存限制一致。
7. 只提供 Linux 格式附加样例文件。

8. 禁止在源代码中改变编译器参数（如使用 #pragma 命令），禁止使用系统结构相关指令（如内联汇编）和其他可能造成不公平的方法。

货币代码（coin）

【题目描述】

在一个科技日新月异的时代，小蓝是国家级铸币局安全实验室的首席技术官。这个实验室的使命是设计和实施国家货币的防伪技术，以保护国家经济免受伪造货币的侵害。随着数字货币的兴起和全球经济的数字化转型，传统的物理货币安全措施正面临着前所未有的挑战。为了应对这些挑战，小蓝和他的团队一直在寻找一种新的、更加安全的防伪解决方案。

他们提出了一个创新的想法：利用一种特殊的字符串编码系统来为每一枚硬币创建一个独一无二的身份标识。这个系统的核心是一个被称为“基础字符串” B 的字符序列，以及一个与之配套的数字序列 S。这个数字序列 S 将决定从基础字符串 B 中选择哪些字符来形成最终的安全代码。具体来说，如果序列 S 中的第 i 个元素是 S_i，那么生成的安全代码中的第 i 个字符就是基础字符串 B 中的第 S_i 个字符（即 $B[S_i]$）。这样，生成的安全代码的长度将与数字序列 S 的长度相等。

这个数字序列 S 的生成方式类似于著名的斐波那契数列，但又有所不同。斐波那契数列以两个 1 开始，而小蓝的团队使用的序列则以两个不同的数字开始，这两个数字分别作为序列的第一和第二个元素，且第一个数字总是小于第二个数字(这两个数字不能超过 B 的长度 n)。序列中的第三个数字是前两个数字的和，第四个数字是第二个和第三个数字的和，以此类推。这个生成过程会在新生成的数字超过基础字符串 B 的长度时停止，以确保所有生成的数字都在基础字符串 B 的范围内。显然地，当我们选择不同的数字作为起始的两个数时，声称的安全代码是不同的。

小蓝意识到，尽管这种方法在理论上可以生成大量的安全代码，但在实际操作中可能会遇到重复代码的问题。为了避免这种情况，他需要计算出使用这种方法实际能生成多少种不同的安全代码，即，所有可能采用的起始数字所生成的所有安全代码中有多少种不同的字符串。

由于小蓝需要集中精力领导他的团队推进这个创新的安全项目，他希望你能帮助他解决这个复杂的数学问题，以便他的团队能够继续确保国家货币的安全性和可靠性。

【输入格式】

从文件 ***coin.in*** 中读入数据。

第一行输入基础字符串 B。基础字符串由数字、小写和大写拉丁字母组成。

【输出格式】

输出到文件 ***coin.out*** 中。

输出一行一个整数，表示能够生成的不同安全代码的数量。

【样例 1 输入】

```
abba
```

【样例 1 输出】

```
5
```

【样例 1 解释】

总共能生成 5 种代码，他们分别是：

- 1 2 3 -> abb
- 1 3 4 -> aba
- 1 4 -> aa
- 2 3 -> bb
- 3 4 -> ba

【样例 2 输入】

```
FAKE0123456789original
```

【样例 2 输出】

```
217
```

【数据范围】

记 n 为字符串 B 的长度。

对于所有测试数据保证：$n \leq 1000$。

测试点编号	$n \leq$	特殊性质
$1 \sim 4$	30	无
$5 \sim 8$	100	无
$9 \sim 12$	1000	字符串 B 全部由字符 a 组成
13		字符串 B 仅包含小写字母
14		字符串 B 仅包含大写字母
15		字符串 B 仅包含数字
$16 \sim 20$		无

Sumplete 游戏（game）

【题目描述】

Sumplete 是一种类似于数独的逻辑谜题。它因在 ChatGPT 的帮助下开发而闻名。

Sumplete 谜题由一个正方形网格组成，每个单元格中包含一个整数。每一行和每一列也分配了一个整数“提示”。玩家必须划掉网格中的一些数字，使得每行和每列中未划掉的数字之和等于相应的提示。下面提供了一个图示来说明这一点。

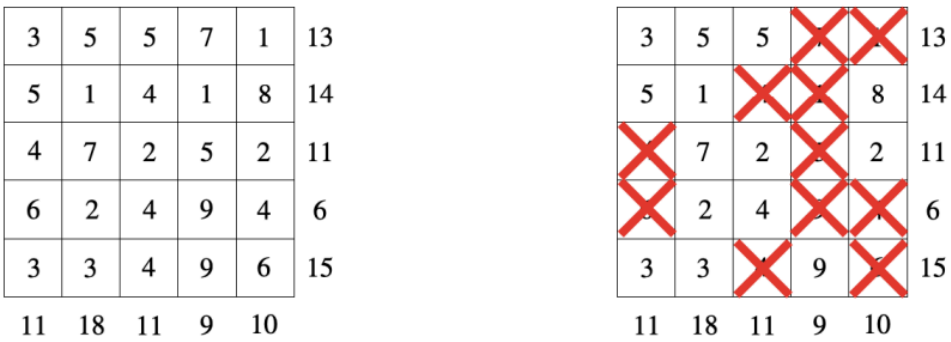


图 1　一个 5 × 5 Sumplete 谜题的例子（左）及其解答（右）。

最近，在 2023 年 9 月 15 日，Suthee Ruangwises 发表了一篇名为《Sumplete is Hard, Even with Two Different Numbers》的论文，上传到了 arxiv。该论文表明，如果我们允许 Sumplete 谜题的网格是矩形的，那么决定给定 Sumplete 谜题是否可解这个问题是 NP-完全的，即使网格中只包含两个不同的数字 1 和 3。

小蓝对这个结果很不满意。他坚持认为一定存在一些容易解决的 Sumplete 谜题。现在他提供了一个 Sumplete 谜题，网格中只包含两个不同的数字 -1 和 1。与论文中所提及的相同，对于给出的 Sumplete 谜题，你只需要判断它是否是可解的。你能帮他解决这个谜题吗？

【输入格式】

从文件 ***game.in*** 中读入数据。

第一行包含一个整数 T，表示数据组数。

对于每组数据，第一行包含一个整数 n，表示 Sumplete 谜题的网格高度和宽度。

接下来的 n 行中，第 i 行包含一个长度为 n 的字符串 s_i，由字符 ‘+’ 和 ‘-’ 组成，其中：

- 如果 $s_{i,j}$=+,则 $a_{i,j} = 1$
- 如果 $s_{i,j}$=-，则 $a_{i,j} = -1$

然后，输入包含一行 n 个整数 $r_1, r_2, ..., r_n$，表示第 i 行的“提示”。

最后，输入包含一行 n 个整数 $c_1, c_2, ..., c_n$，表示第 i 列的“提示”。

【输出格式】

输出到文件 ***game.out*** 中。

对于每组数据，输出一行一个字符串 Yes 或 No。如果给定的 Sumplete 谜题没有解，则在输出 No。否则，输出 Yes。

【样例 1 输入】

```
3
3
+-+
-++
+-+
1 1 1
1 -1 3
3
---
-++
+++
-2 -1 0
-2 -1 0
3
+-+
-++
++-
1 0 2
2 2 -1
```

【样例 1 输出】

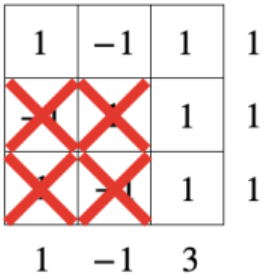
```
Yes
Yes
No
```

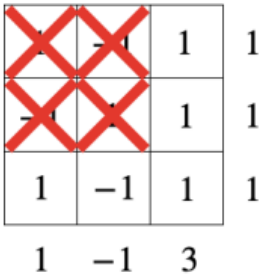
【样例 1 解释】

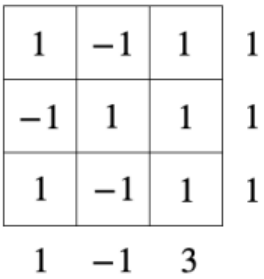
第一组数据对应于以下 3×3 的数独拼图：

1	−1	1	1
−1	1	1	1
1	−1	1	1
1	−1	3	

他有如下三种可能的解（第三个解对应于在网格中不划掉任何数字）：







【数据范围】

对于所有测试数据保证：$1 \le T \le 1000, 1 \le n \le 4000, 1 \le \sum n^2 \le 2 \times 10^7, -n \le r_i, c_i \le n$.

测试点编号	$\sum n^2 \le$	$n \le$	特殊性质
$1 \sim 2$	160	4	无
$3 \sim 4$	10000	100	无
$5 \sim 7$	2×10^7	4000	A
$8 \sim 10$			无

特殊性质 A: 在一组数据中，Sumplete 谜题网格要么全为 +, 要么全为 -。

高速公路（road）

【题目描述】

一条全新的联通伯伦希尔和休伯利安的三 A 级收费高速公路即将开放。当然，这条高速公路是双向的，但为了这个问题，我们只考虑从伯伦希尔到休伯利安的方向。

这条公路的长度为 L，并且有 n 个特殊点。所有特殊点都位于公路沿线的一些不同的整数坐标上，其中第一个位于位置 0，最后一个位于位置 L。每个特殊点要么是入口，要么是出口。当然，位于位置 0 的特殊点是入口，位于位置 L 的特殊点是出口。

使用这条高速公路的每辆车都将配备一个特殊的设备，称为 ETC 电子标签。公路的所有者将在公路沿线的某些位置设置收费亭，与过往车辆的 ETC 电子标签进行通信。公路沿线可以建造两种类型的收费亭：

1. 入口或出口收费亭。它们只能建在包含入口或出口的位置。这样的收费亭建造成本为 a 元，并与使用相应入口或出口的所有车辆的 ETC 电子标签通信。
2. 公路收费亭。它们只能建在半整数位置，即 $k+0.5$ 点(k 可取 0 到 $L-1$ 之间的任意一个整数，包含 0 和 $L-1$)。这样的收费亭建造成本为 b 元，并与所有沿高速公路行驶且经过此收费亭的车辆的 ETC 电子标签通信。

为了妥善收取所有使用高速公路的车辆的费用，需要能够识别每辆车的确切入口位置和确切出口位置。正式地说，你需要建造一组收费亭以满足以下要求：

1. 每辆在某个入口进入公路，向前行驶然后从某个出口离开的车辆，应该至少经过一个收费亭，这样我们就可以识别这辆车使用了高速公路。
2. 对于每辆在某个入口进入公路，向前行驶然后从某个出口离开的车辆，我们应该能够根据这辆车的 ETC 电子标签与收费亭之间的通信信息，准确地识别入口和出口的位置。请注意，由于复杂的收费制度，我们感兴趣的是入口和出口的确切位置，而不仅仅是沿高速公路行驶的距离。
3. 建造所有收费亭的总成本应尽可能低。

你只需要输出收费亭的总成本就可以了。

【输入格式】

从文件 ***road.in*** 中读入数据。

输入的第一行包含一个整数 T 表示数据组数。

对于每组数据，第一行输入四个整数 n, L, a 和 b，分别是公路沿线特殊点的数量、公路的长度、在入口或出口处建一个收费亭的成本，以及在公路沿线建一个收费亭的成本。

接下来的 n 行中，每一行都包含一个字符 c_i 和一个整数 x_i，中间用一个空格隔开。字符 c_i 表示特殊点类型，可以是 E（入口）或 T（出口），而 x_i 则表示该点在道

路上的位置。保证所有的 x 都是不同的，并且在 0 位置有一个入口，在 L 位置有一个出口。

【输出格式】

输出到文件 ***road.out*** 中。

对于每组数据，输出在一行一个整数，表示合法的建造收费亭方案的最低总成本。

【样例 1 输入】

```
4
2 10 1 1
E 0
T 10
2 10 3 2
E 0
T 10
5 40 5 6
E 0
T 40
T 20
E 30
E 10
10 9 5 6
T 5
E 6
T 7
E 8
T 9
E 0
T 1
E 2
T 3
E 4
```

【样例 1 输出】

```
1
2
15
28
```

【数据范围】

对于所有测试数据保证：$1 \le t \le 1000, 2 \le n \le \sum n \le 5 \times 10^5, 1 \le L, a, b \le 10^9$.

测试点编号	$\sum n \le$	$n \le$	特殊性质
1	50	5	无
2	500	50	
$3 \sim 4$	10000	1000	
5	5×10^5	5×10^5	$a = 1, b = 10^9$
6			$a = 10^9, b = 1$
$7 \sim 10$			无

加数（add）

【题目描述】

小蓝是一个魔法建筑师，他能让高楼在平地上拔地而起。现在，小蓝有一个整数序列 $x_1, x_2, ..., x_n$，表示 n 栋楼的高度。最初，这里还是一片平地，因此所有 x_i 的值都为 0。然后，他可以执行以下操作任意次数：

- 选择一个整数 k（$1 \le k \le n$）。然后，为 $1 \le i \le k$ 的每个 x_i 加一，即让前 k 栋楼的高度提升 1。这样的操作成本是 k。
- 选择一个整数 k（$1 \le k \le n$）。然后，为 $n-k+1 \le i \le n$ 的每个 x_i 加一，即让后 k 栋楼的高度提升 1。这样的操作成本是 k。

小绿希望小蓝建成的楼能满足实际的高度需求。具体地，他也有 n 个整数 $y_1, y_2, ..., y_n$。他希望小蓝执行一系列操作后，使得条件 $x_i \ge y_i$ 对所有 $1 \le i \le n$ 成立。

这项任务对小蓝来说太简单了。所以他想知道达到目标所需的最低成本是多少。因此，你的任务是编写一个程序来计算最低成本。

【输入格式】

从文件 ***add.in*** 中读入数据。

输入的第一行包含一个整数 T 表示数据组数。

对于每组数据，第一行输入一个整数 n 表示整数序列的大小。

第二行输入一行 n 个整数 $y_1, y_2, ..., y_n$，意义如题所示。

【输出格式】

输出到文件 ***add.out*** 中。

对于每组数据，输出一行一个整数，表示达到目标所需的最低成本。

【样例 1 输入】

```
3
5
1 1 1 1 1
10
0 0 0 0 1000000000 0 0 0 0 0
13
1 1 4 5 1 4 1 9 1 9 8 1 0
```

【样例 1 输出】

```
5
5000000000
76
```

【数据范围】

对于所有测试数据保证：$1 \le T \le 10^5, 1 \le n \le \sum n \le 10^6, 0 \le y_i \le 10^9$。

测试点编号	$\sum n \le$	$n \le$	特殊性质
$1 \sim 2$	20	4	$y_i \le 5$
$3 \sim 4$	200	50	无
$5 \sim 6$	5×10^4	5×10^4	$y_i \le 100$
$7 \sim 8$	6000	2000	无
$9 \sim 11$	10^6	10^6	A
$12 \sim 13$			B
$14 \sim 20$			无

特殊性质 A: y_i 的取值种类不超过 2 种。

特殊性质 B: 序列 y 存在大小关系 $y_1 \ge \sum_{i=1}^{n-1} \max(0, y_i - y_{i+1})$