

2024 洛谷网校基础-提高衔接计划

模拟赛

第一场

时间：2024 年 3 月 9 日 08:30 ~ 12:00

题目名称	序列	洛谷波特	购买	航线规划
题目类型	传统型	传统型	传统型	传统型
可执行文件名	sequence.exe	bot.exe	buy.exe	flight.exe
输入文件名	sequence.in	bot.in	buy.in	flight.in
输出文件名	sequence.out	bot.out	buy.out	flight.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
测试点数目	20	10	20	25
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	sequence.cpp	bot.cpp	buy.cpp	flight.cpp
-----------	--------------	---------	---------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=536870912
-----------	--------------------------------------

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为 0。
3. 选手源程序必须存放于题目指定的目录（子文件夹）下。
4. 程序可使用的栈空间大小与该题内存空间限制一致。
5. 若无特殊说明，每道题的源代码大小限制为 100KB，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔，结果比较方式为全文比较（忽略行末空格、文末回车）。
6. 统一评测时采用的机器配置为 Intel(R) Xeon(R) Platinum 8369B CPU @ 2.70GHz，内存 16 GB。上述时限以此配置为准。
7. 评测在 Linux 下进行。数据与附加样例文件均为 Linux 格式的。
8. 选手不得在源程序中使用内嵌汇编，也禁止以任何方式修改程序的编译选项。

序列 (sequence)

【题目描述】

给定 n 个整数 $a_1, a_2, \dots, a_n$ 。请你从中挑选尽可能多的数，使得这些数的极差不超过 k 。

一组数的极差定义为这组数的最大值与最小值的差。

【输入格式】

从文件 `sequence.in` 中读入数据。

第一行是两个整数，依次表示数字个数 n 和极差限制 k 。

第二行有 n 个整数，依次表示 $a_1, a_2, \dots, a_n$ 。

【输出格式】

输出到文件 `sequence.out` 中。

输出一行一个整数表示能挑出的数字数量。

【样例 1 输入】

```
1 5 7
2 9 2 1 6 3
```

【样例 1 输出】

```
1 4
```

【样例 1 解释】

可以选择 9, 2, 6, 3 这四个数。

【样例 2】

见选手目录下的 `sequence/sequence2.in` 与 `sequence/sequence2.ans`。

【子任务】

- 对于 25% 的数据, $n \leq 20$ 。
- 对于 40% 的数据, $n \leq 300$, $a_i \leq n$ 。
- 对于 55% 的数据, $n \leq 300$ 。
- 对于 80% 的数据, $n \leq 2000$ 。
- 另有 5% 的数据, $k = \max\{a_i\}$ 。
- 对于 100% 的数据, $1 \leq n \leq 10^5$, $1 \leq k, a_i \leq 10^9$ 。

洛谷波特 (bot)

【题目背景】

洛谷研发了新一代机器人『洛谷波特』。现在，扶苏作为波特的开发者，正在对波特进行测试。

【题目描述】

波特被放在了一个无限大的平面中，初始时位于原点 $(0,0)$ 。

扶苏会给波特发送一个只含字符 U、D、R、L 的字符串。字符串的每一个字母都是一次移动指令，四个字母的含义如下：

- U：机器人向上移动一格，假设机器人当前坐标为 (x,y) ，则移动到 $(x,y+1)$ 。
- D：机器人向下移动一格，假设机器人当前坐标为 (x,y) ，则移动到 $(x,y-1)$ 。
- R：机器人向右移动一格，假设机器人当前坐标为 (x,y) ，则移动到 $(x+1,y)$ 。
- L：机器人向左移动一格，假设机器人当前坐标为 (x,y) ，则移动到 $(x-1,y)$ 。

平面上有一些障碍物，障碍物均位于整点坐标上。如果机器人的下一个移动会移动到障碍物上，那么它将忽略这个移动指令，并尝试执行接下来的移动指令。

当指令执行结束后，机器人将会停止运动。

遗憾的是，因为权限不足，扶苏并不知道平面上有哪些位置有障碍物。因此，她想向你询问，对于障碍物所有可能的分布，机器人一共可能在哪些位置停下。

【输入格式】

从文件 `bot.in` 中读入数据。

输入的第一行是一个整数，表示字符串的长度 n 。

第二行有一个长度为 n 的字符串 s ，表示给出的指令串。

【输出格式】

输出到文件 `bot.out` 中。

输出的第一行是一个整数 k ，表示机器人可能停下的位置。

接下来 k 行，每行两个整数 x,y ，用空格隔开，表示一个机器人可能停下的坐标。

注意，你应该按 x 从小到大的顺序输出这些坐标。当两个坐标的 x 相同时，先输出 y 较小的坐标。

【样例 1 输入】

```
1 2
2 UR
```

【样例 1 输出】

```
1 4
2 0 0
3 0 1
4 1 0
5 1 1
```

【样例 2 输入】

```
1 4
2 URDL
```

【样例 2 输出】

```
1 6
2 -1 -1
3 -1 0
4 0 -1
5 0 0
6 0 1
7 1 -1
```

【样例 3】

见选手目录下的 *bot/bot3.in* 与 *bot/bot3.ans*。

【子任务】

对于全部的测试点，保证 $1 \leq n \leq 20$ ， s 中仅含字母 U、D、R、L。

测试点	$n=$	特殊性质
1 ~ 3	3	无
4	10	s 中仅含字符 R
5		s 中仅含字符 L 和 R
6		s 中仅含字符 U 和 D
7, 8		无
9, 10	20	

购买 (buy)

【题目描述】

扶苏来到了一个神秘的超市。这个超市有 n 件商品，第 i 件商品有两个价格参数 a_i, b_i 。

扶苏将会从 n 件商品中挑选恰好 k 件商品。在付款时，扶苏需要支付的金额是这 k 件商品参数 a_i 之和加上参数 b_i 的最大值。

形式化的，假设扶苏选择的商品的下标是 $p_1, p_2, \dots, p_k (1 \leq p_1 < p_2 < \dots < p_k \leq n)$ ，她需要支付的金额是：

$$\sum_{i=1}^k a_{p_i} + \max_{i=1}^k \{b_{p_i}\}$$

只有购买 k 件商品，她才能从这个商店离开。扶苏想让你求出：她最少花费多少元才能购买恰好 k 件商品。

【输入格式】

从文件 *buy.in* 中读入数据。

本题单个测试点内有多组测试数据。输入的第一行是一个整数，表示数据组数 T 。对每组数据，按如下格式输入：

第一行是两个整数，依次表示商品个数 n 和应挑选的商品数量 k 。

第二行有 n 个整数，第 i 个整数表示 a_i 。

第三行有 n 个整数，第 i 个整数表示 b_i 。

【输出格式】

输出到文件 *buy.out* 中。

对于每组测试数据，输出一行一个整数表示答案。

【样例 1 输入】

```
1 3
2 3 2
3 1 2 3
4 3 2 1
5 5 3
6 1 1 1 2 3
7 5 4 3 2 1
```

```
8 3 2
9 1 2 3
10 100 1 1
```

【样例 1 输出】

```
1 6
2 8
3 6
```

【样例 1 解释】

- 对第一组数据, 选择第 1, 2 件商品。花费为 $(1 + 2) + \max\{3, 2\} = 3 + 3 = 6$ 。
- 对第二组数据, 选择第 2, 3, 4 件商品, 花费为 $(1 + 1 + 2) + \max\{4, 3, 2\} = 4 + 4 = 8$ 。
- 对第三组数据, 选择第 2, 3 件商品, 花费为 $(2 + 3) + \max\{1, 1\} = 5 + 1 = 6$ 。

【样例 2】

见选手目录下的 *buy/buy2.in* 与 *buy/buy2.ans*。

【子任务】

- 对于 15% 的数据, $k = 1$ 。
- 对于 35% 的数据, $n \leq 20$ 。
- 对于 65% 的数据, $n \leq 10^3$ 。
- 对于 85% 的数据, $n \leq 10^4$ 。
- 对于 100% 的数据, $1 \leq k \leq n \leq 10^6$, $1 \leq a_i, b_i \leq 10^{12}$, $1 \leq T \leq 3$ 。

【提示】

请注意大量数据读入对程序效率造成的影响, 选择合适的读入方式, 避免超时。

航线规划 (flight)

【题目描述】

扶苏是洛谷王国的航空局长，她掌管着洛谷境内一切民航飞机的航线安排。

洛谷王国共有 n 座城市，从 1 至 n 编号。任意两座城市之间都有一条双向航线。但不一定每条航线都有航班经过。

初始时，每条航线上都没有航班。在接下来 m 天里，扶苏会根据对人们出行需求的预测，在若干天内向一些航线上添加航班，并做出一些查询。

每次添加航班的格式是： $u\ v\ x\ y$ ，表示在城市 u 和城市 v 的航线上添加 x 个航班，这条命令会在 y 天后取消。即给从增加命令下达当天到下达后第 $y - 1$ 天这 y 天里 u 到 v 的航线添加 x 个航班。在第 y 天不再添加 x 个航班。可能会有多个命令影响同一天的航班数，这些命令可以叠加生效。

对于一个城市 u ，称 u 的“主要出行目的地”是当前时刻与 u 之间的航线上航班数最多的城市（如果存在多个这样的城市，取编号最小的）；特别的，如果 u 与任何城市之间的航线上航班数均为 0，称 u 是一个“封闭城市”，且它没有“主要出行目的地”；如果两座城市互为“主要出行目的地”，称他们是“交流城市对”。

每天开始时，扶苏会先下达若干条添加航班的命令。在处理完这些命令后，你会收到若干个查询某个城市的“主要出行目的地”的请求。此外，你还可能希望求出此时的“封闭城市”和“交流城市对”各有多少。

请你帮助处理扶苏下达的添加航班的指令，并回答她的询问。

【输入格式】

从文件 `flight.in` 中读入数据。

第一行是两个整数，依次表示城市数量 n 和天数 m 。

接下来 $3m$ 行，每三行一组描述一天中的事件，格式如下：

- 第一行：首先是一个整数 k_i ，表示当天下达添加航班的命令数。接下来有 $4k_i$ 个整数，依次表示这 k_i 条命令。格式见【题目描述】。
- 第二行：首先是一个整数 l_i ，表示当天查询“主要出行目的地”的数量。接下来 l_i 个整数，依次表示每次查询的对象。
- 第三行：只有两个整数 p_i, q_i ，取值均为 0 或 1。分别表示当天是否需要查询“封闭城市”和“交流城市对”的数量。如果 $p_i = 1$ ，表示查询“封闭城市”的数量，如果 $p_i = 0$ ，表示不需要查询； q_i 的含义同理。

【输出格式】

输出到文件 `flight.out` 中。

对每个查询，输出一行一个整数表示答案。

输出应该按照查询所在天数的顺序输出。对于同一天内的查询，先按照被查询的顺序输出每个被查询对象的”主要出行目的地“数量，再依次输出”封闭城市对“的数量（如果不需要输出则不输出）和”交流城市对“的数量（如果不需要输出则不输出）。

如果某个查询“主要出行目的地”的城市是“封闭城市”，认为查询结果为 0。

【样例 1 输入】

```
1 3 3
2 2 1 2 2 3 1 3 3 2
3 1 1
4 0 0
5 1 2 3 3 1
6 2 1 2
7 0 1
8 0
9 2 1 3
10 1 1
```

【样例 1 输出】

```
1 3
2 3
3 3
4 1
5 2
6 0
7 1
8 1
```

【样例 2】

见选手目录下的 *flight/flight2.in* 与 *flight/flight2.ans*。

这个样例满足 $p_i = q_i = 0$ 。

【样例 3】

见选手目录下的 *flight/flight3.in* 与 *flight/flight3.ans*。

这个样例满足 $l_i = q_i = 0$ 。

【样例 4】

见选手目录下的 *flight/flight4.in* 与 *flight/flight4.ans*。

这个样例满足 $l_i = 0$ 。

【样例 5】

见选手目录下的 *flight/flight5.in* 与 *flight/flight5.ans*。

这个样例满足 $k_2 = k_3 = \dots = k_m = 0$ ，且所有添加命令的 $y = m$ 。

【样例 6】

见选手目录下的 *flight/flight6.in* 与 *flight/flight6.ans*。

【子任务】

以下设 $A = \sum_{i=1}^m k_i$ ， $B = \sum_{i=1}^m l_i$ 。

- 对 32% 的数据， $n, m \leq 1000$ ， $A, B \leq 2000$ 。
- 另有 8% 的数据， $p_i = q_i = 0$ 。
- 另有 8% 的数据， $l_i = q_i = 0$ 。
- 另有 12% 的数据， $l_i = 0$ 。
- 另有 8% 的数据， $k_1 = A$ ，且所有添加命令的 $y = m$ 。
- 另有 12% 的数据，所有添加命令的 $y = m$ 。
- 对 100% 的数据， $1 \leq n, m \leq 10^5$ ， $1 \leq A, B \leq 2 \times 10^5$ ， $1 \leq u, v \leq n$ ， $1 \leq x \leq 10^9$ ， $1 \leq y \leq m$ ， $u \neq v$ 。