

NOIP 模拟赛

WZMS

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	背包	袋鼠	接雨水	排列问题
子目录名	knapsack	kangaroo	rain	puzzle
可执行文件名	knapsack	kangaroo	rain	puzzle
输入文件名	knapsack.in	kangaroo.in	rain.in	puzzle.in
输出文件名	knapsack.out	kangaroo.out	rain.out	puzzle.out
每个测试点时限	1000ms	1000ms	5000ms	4000ms
内存上限	1G	512M	1G	512M
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5
附加样例文件	有	有	有	有
结果比较方式	全文比较	全文比较	全文比较	Special Judge
题目类型	传统	传统	传统	传统

二、提交源程序程序名

对于 C++ 语言	knapsack.cpp	kangaroo.cpp	rain.cpp	puzzle.cpp
-----------	--------------	--------------	----------	------------

三、优化开关

对于 C++ 语言	-O2 -std=c++17
-----------	----------------

注意事项：

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/C++ 中函数 main() 的返回类型必须是 int，程序正常结束时的返回值必须是 0；
3. 保证各个题目时间限制至少为标程运行时间的两倍；
4. 题目难度不一定升序排序；
5. 每道题目需单独建立对应子文件夹。

1 背包

(knapsack.cpp)

1.1 问题描述

小 A 是一个缺乏经验的商人，最近开了一家名为「Queen's Organic Jewelry」的珠宝店。这家珠宝店里有 n 颗宝石，其中第 i 颗宝石的价格为 w_i 美元，具有美丽值 v_i 。在拜访珠宝店之前，你手上有 W 美元，你计划用这些钱购买尽可能总美丽值最高的宝石。

有趣的是，小 A 的珠宝店今天正在进行促销活动。任何到店的顾客可以选择任意 k 颗宝石并免费带走！有了这个机会，你迫切想知道在最优策略下可以获得的宝石的最大总美丽值。

请注意，店里每颗宝石只有一件，因此你不能获得同一颗宝石多次。同时注意，你不一定要花光所有的钱。

1.2 输入

第一行包含三个整数 n , W 和 k ，分别表示珠宝店中的宝石总数、你拥有的金额以及你可以免费带走的宝石数量。满足 $1 \leq n \leq 5 \times 10^3$, $1 \leq W \leq 10^4$, $0 \leq k \leq n$ 。

接下来的 n 行中，第 i 行包含两个整数 w_i 和 v_i ，分别表示第 i 颗宝石的价格和美丽值。满足 $1 \leq w_i \leq W$, $1 \leq v_i \leq 10^9$ 。

1.3 输出

输出一行，包含一个整数，表示可以获得的最大总美丽值。

1.4 输入输出样例 1

1.4.1 输入样例

```
4 10 1
9 10
10 1
3 5
5 20
```

1.4.2 输出样例

35

1.5 输入输出样例 2

1.5.1 输入样例

```
5 13 2
```

5 16
5 28
7 44
8 15
8 41

1.5.2 输出样例

129

1.6 输入输出样例 3

见下发文件中的 knapsack.in 和 knapsack.out。

1.7 约定和数据范围

对所有测试点数据，保证 $1 \leq n \leq 5 \times 10^3$, $1 \leq W \leq 10^4$, $0 \leq k \leq n$, $1 \leq w_i \leq W$, $1 \leq v_i \leq 10^9$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$W \leq$	$k \leq$
1 ~ 4	18	18	18
5 ~ 7	100	100	100
8 ~ 10	500	500	500
11 ~ 12	5×10^3	10^4	2
13 ~ 14	5×10^3	10^3	$\min(n, 5 \times 10^3)$
15 ~ 17	5×10^3	5×10^3	
18 ~ 20	5×10^3	10^4	

2 袋鼠

(kangaroo.cpp)

2.1 问题描述

给定一个 n 行 m 列的网格。每个单元格要么是一个洞，要么是空的。每个空单元格上恰好站着一只袋鼠。

同样地，通过按键盘上的按钮 U、D、L、R 可以控制袋鼠。所有袋鼠将根据按下的按钮同时移动。具体来说，对于位于第 i 行，第 j 列的任何袋鼠（用 (i, j) 表示）：

1. 按下按钮 U：袋鼠会移动到 $(i - 1, j)$ 。
2. 按下按钮 D：袋鼠会移动到 $(i + 1, j)$ 。
3. 按下按钮 L：袋鼠会移动到 $(i, j - 1)$ 。
4. 按下按钮 R：袋鼠会移动到 $(i, j + 1)$ 。

如果袋鼠移动到一个洞上或移出网格，它将从网格中移除。如果在执行一系列操作（可能是空的序列）后，网格上仅剩下一只袋鼠，那么该袋鼠成为获胜者。

问题是：对于每只袋鼠，确定是否存在一种操作序列使它成为获胜者。输出可能成为获胜者的袋鼠总数。

2.2 输入

有多个测试用例。输入的第一行包含一个整数 T ，表示测试用例的数量。对于每个测试用例：

第一行包含两个整数 n 和 m ($1 \leq n, m \leq 10^3$, $1 \leq n \times m \leq 10^3$)，表示网格的行数和列数。

接下来的 n 行中，第 i 行包含一个长度为 m 的字符串 $s_{i,1}s_{i,2}\dots s_{i,m}$ ，其中每个字符要么是.（表示空格，ASCII 码：46），要么是 O（表示洞，大写字母，ASCII 码：79）。如果 $s_{i,j}$ 是.，则网格 (i, j) 是空的；如果 $s_{i,j}$ 是 O，则网格 (i, j) 是一个洞。

保证所有测试用例中 $n \times m$ 的总和不超过 5×10^3 。

2.3 输出

对于每个测试用例，输出一行，包含一个整数，表示存在某种操作序列可以使其成为获胜者的袋鼠数量。

2.4 输入输出样例 1

2.4.1 输入样例

```

2 5
.OO..
O..O.
1 3
O.O
1 3
.O.
2 3
OOO
OOO

```

2.4.2 输出样例

```

3
1
0
0

```

2.5 输入输出样例 4

见下发文件中的 `root.in` 和 `root.out`。

2.6 约定和数据范围

对所有测试点数据，保证 $1 \leq n, m \leq 10^3$ ， $1 \leq n \times m \leq 10^3$ 。

对于所有测试点数据，保证 $n \times m$ 的总和不超过 5×10^3 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$	$n \times m \leq$	$\sum n \times m \leq$
1 ~ 4	100	100	100	500
5 ~ 10	500	500	500	2500
11 ~ 15	750	750	750	3750
16 ~ 20	10^3	10^3	10^3	5×10^3

2.7 样例解释

以下是样例测试用例的解释。我们用“W”表示最终成为获胜者的袋鼠，用“K”表示其他袋鼠。

对于第一个样例测试用例，最初位于 $(1, 4)$ 、 $(1, 5)$ 和 $(2, 5)$ 的袋鼠可能成为获胜者。可能的操作序列如下所示：

K	O	O	W	K
O	K	K	O	K

.	O	O	.	W
O	.	K	O	.

K	O	O	K	W
O	K	K	O	K

.	O	O	.	.
O	.	.	O	W

K	O	O	K	K
O	K	K	O	W

.	O	O	.	W
O	.	.	O	.

对于第二个样例测试用例，由于只有一只袋鼠，它无需任何操作即可成为获胜者。

对于第三个样例测试用例，由于任何操作都会同时移除两只袋鼠，因此没有可能的获胜者。

对于第四个样例测试用例，由于没有袋鼠，因此也没有可能的获胜者。

3 接雨水

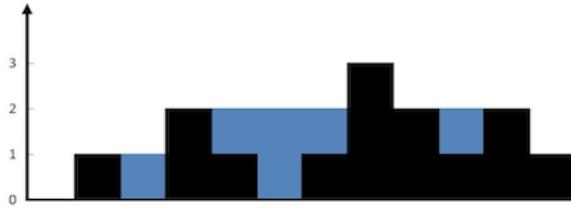
(rain.cpp)

3.1 问题描述

给定一个柱状图，它由长度为 n 的整数序列 a_1, a_2, \dots, a_n 表示。对于从左到右的第 i 根柱子，它的高度为 a_i ，宽度为 1。

我们将对柱状图进行 q 次修改。第 i 次修改可以用一对整数 (x_i, v_i) 表示，表示我们将第 x_i 根柱子的高度增加 v_i 。

在每次修改后，回答以下查询：计算在大雨倾盆而下时，这个柱状图能够积蓄多少水，并填满所有的凹槽。



更正式地说，给定一个长度为 n 的整数序列 a_1, a_2, \dots, a_n ，第 i 次修改将增加 a_{x_i} 的高度 v_i 。在每次修改后，回答以下查询：设 $f_i = \max(a_1, a_2, \dots, a_i)$ 和 $g_i = \max(a_i, a_{i+1}, \dots, a_n)$ ，计算：

$$\sum_{i=1}^n (\min(f_i, g_i) - a_i)$$

3.2 输入

有多个测试用例。输入的第一行包含一个整数 T ，表示测试用例的数量。对于每个测试用例：

第一行包含一个整数 n ($1 \leq n \leq 10^5$)，表示柱状图中的柱子数量。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$)，其中 a_i 表示第 i 根柱子的初始高度。

第三行包含一个整数 q ($1 \leq q \leq 10^5$)，表示修改的数量。

接下来的 q 行中，第 i 行包含两个整数 x_i 和 v_i ($1 \leq x_i \leq n$, $1 \leq v_i \leq 10^6$)，表示第 i 次修改将第 x_i 根柱子的高度增加 v_i 。

保证所有测试用例中 n 和 q 的总和不超过 10^6 。

3.3 输出

对于每次修改，输出一行，包含一个整数，表示这个柱状图能够积蓄的雨水总量。

3.4 输入输出样例 1

3.4.1 输入样例

```

2
6
1 2 3 4 5 6
2
1 2
3 3
5
100 10 1 10 100
1
3 100

```

3.4.2 输出样例

```

1
4
180

```

3.5 约定和数据范围

对所有测试点数据，保证 $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$, $1 \leq a_i \leq 10^6$, $1 \leq x \leq n$, $1 \leq v_i \leq 10^6$ 。
保证所有测试用例中 n 和 q 的总和不超过 10^6 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$q \leq$	$\sum n + q \leq$	特殊性质
1 ~ 6	5000	5000	50000	无
7 ~ 10	10^5	10^5	10^6	特殊性质 A
11 ~ 14	10^4	10^4	10^5	无
15 ~ 16	5×10^4	5×10^4	5×10^5	
17 ~ 20	10^5	10^5	10^6	

特殊性质 A：保证任意时刻，柱状图的高度满足单谷的性质。存在一个柱子，它的左边的柱子形成了一个单调递减序列，右边的柱子形成了一个单调递增序列。

更形式化地来讲，单谷序列是指 $\exists i (1 \leq i \leq n)$, 满足 $a_1 \geq a_2 \geq \dots \geq a_i, a_i \leq a_{i+1} \leq \dots \leq a_n$ 。

4 排列问题

(puzzle.cpp)

4.1 问题描述

小 A 正在解一个谜题。这个谜题的关键是一个包含数字 1 到 n 的排列。排列中某些位置的值已经固定，而 A 需要将数字填入剩余的位置。

此外，小 A 还收集了 m 个关于排列的额外要求。假设解表示为 p_1, p_2, \dots, p_n ，每个线索都是一对索引 (u_i, v_i) ，这意味着在解中应该满足 $p_{u_i} < p_{v_i}$ 。

小 A 想知道是否可以同时满足所有要求。编写程序以找出有效的解，如果存在的话。

4.2 输入

输入的第一行包含测试用例的数量 T ($1 \leq T \leq 2 \times 10^4$)。

对于每个测试用例：

第一行包含两个整数 n, m ($2 \leq n \leq 2 \times 10^5$, $1 \leq m \leq 5 \times 10^5$)。

第二行包含 n 个整数 p_1, p_2, \dots, p_n ($0 \leq p_i \leq n$)。如果 $1 \leq p_i \leq n$ ，则位置 i 的值是固定的 p_i ，否则你需要确定位置 i 的值。保证对于 $1 \leq i < j \leq n$ ，如果 $p_i > 0$ 且 $p_j > 0$ ，则 $p_i \neq p_j$ 。

接下来的 m 行中，每行包含两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n$)，表示线索。保证线索之间没有矛盾。正式来说，不存在一系列线索 $(u_{i_1}, v_{i_1}), (u_{i_2}, v_{i_2}), \dots, (u_{i_k}, v_{i_k})$ 使得 $v_{i_j} = u_{i_{j+1}}$ ， $1 \leq j < k$ 且 $v_k = u_{i_1}$ 。

保证所有测试用例中 n 的总和不超过 2×10^5 ， m 的总和不超过 5×10^5 。

4.3 输出

对于每个测试用例：

如果不存在有效解，输出”-1”。

否则，输出一行包含 n 个整数，以空格分隔，表示解。如果有多个解，输出任意一个。

4.4 输入输出样例 1

4.4.1 输入样例

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

3 2
0 3 1
1 2
3 1

4.4.2 输出样例

1 3 2 4
2 3 1

4.5 输入输出样例 2

4.5.1 输入样例

1
4 4
1 4 0 0
1 2
1 3
2 4
3 4

4.5.2 输出样例

-1

4.6 约定和数据范围

对所有测试点数据，保证 $1 \leq n \leq 2 \times 10^5$, $1 \leq m \leq 5 \times 10^5$, $0 \leq p_i \leq n$, $1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $u_i < v_i$, $p_i \neq p_j$ 。

对于所有测试点数据，保证 n 的总和不超过 2×10^5 , m 的总和不超过 5×10^5 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$	$\sum n \leq$	$\sum m \leq$	特殊性质
1 ~ 4	10	25	50	50	无
5 ~ 8	100	250	500	750	
9 ~ 10	2×10^5	5×10^5	2×10^5	5×10^5	特殊性质 A
11 ~ 14	1000	2500	5000	7500	无
15 ~ 16	10^5	25000	2×10^5	5×10^5	
17 ~ 20	2×10^5	5×10^5	2×10^5	5×10^5	

特殊性质 A：保证排列中未确定的元素数量不超过 10，且 $T \leq 5$ 。