

模拟赛

2020 年 11 月 21 日

题目名称	T1 出了个大阴间题	T2 最简单辣快来做	T3 是我的你不要抢	T4 显然也是我整的
题目类型	传统型	传统型	传统型	传统型
目录	repair	satellite	string	graph
可执行文件名	repair	satellite	string	graph
输入文件名	repair.in	satellite.in	string.in	graph.in
输出文件名	repair.out	satellite.out	string.out	graph.out
每个测试点时限	2 秒	2 秒	2 秒	2 秒
内存限制	512 MB	512 MB	512 MB	512 MB
子任务数目	10	20	20	20

提交源程序文件名

对于 C++ 语言	repair.cpp	satellite.cpp	string.cpp	graph.cpp
-----------	------------	---------------	------------	-----------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

注意事项

1. 需要建立子文件夹。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. 结果比较方式为忽略行末空格、文末回车后的全文比较。
4. C/C++ 中函数 main() 的返回值类型必须是 int，值为 0。

T1 出了个大阴问题 (repair)

【题目描述】

因为你是小 ω ，所以你是一个奇妙的能力，能把两个二元组通过一定的代价合并成一个。

具体来说，你可以将 (a_1, b_1) 与 (a_2, b_2) 合并成 (a, b) ，满足以下条件：

$$a = \begin{cases} \max(a_1, a_2) & a_1 \neq a_2 \\ a_1 + 1 & a_1 = a_2 \end{cases}$$

$$b = 2 \max(b_1, b_2) + 1$$

需要的花费刚好是 $ka + b_1 + b_2$ 。

现在你有 n 个二元组 $(a_1, 0), (a_2, 0), \dots, (a_n, 0)$ ，你需要按照一定的顺序将它们全部合并。

你按照一个排列 $p_1, p_2, p_3, \dots, p_n$ 的顺序合并，就是先将 $(a_{p_1}, 0)$ 与 $(a_{p_2}, 0)$ 合并，再将所得结果与 $(a_{p_3}, 0)$ 合并，以此类推。

你希望最后结果 (a, b) 中 a 最大，并希望求出在 a 最大的条件下所有合法排列的合并代价总和。（答案对 $10^9 + 7$ 取模）

【输入格式】

从文件 `repair.in` 中读入数据。

第一行输入两个正整数 n, k 。

第二行输入 n 个正整数 a_i 。

【输出格式】

输出到文件 `repair.out` 中。

共一行输出两个整数，表示最大的 a 与代价总和。

【样例 1 输入】

```
3 1
1 1 2
```

【样例 1 输出】

3 12

【样例 1 解释】

共有两个排列符合条件：1,2,3 和 2,1,3，合并代价均为 6。

【样例 2】

见选手目录下的 *repair/repair2.in* 与 *repair/repair2.ans*。

【数据范围】

对于所有数据，保证 $1 \leq n \leq 18$ ， $1 \leq a_i \leq 10^9$ ， $1 \leq k < 10^9 + 7$ 。

- 对于第 1 ~ 4 组数据，保证 $n \leq 10$ 。
- 对于第 5 ~ 6 组数据，保证 $a_i \leq n \leq 15$ 。
- 对于第 7 ~ 8 组数据，保证 $a_i \leq n$ 。
- 对于第 9 ~ 10 组数据，没有特殊限制。

T2 最简单辣快来做 (satellite)

【题目描述】

Ω 城的布局可以看作一个 $w \times h$ 的网格。在 Ω 城的上空有 n 个卫星，第 i 个卫星的高度在网格 (x_i, y_i) 上空高度为 h_i 的位置。

(在本题中，你不需要思考这些卫星为什么可以同步运转)

你作为 Ω 城的市长小 ω ，需要建造一个通信中心。通信中心的位置如果在 (p, q) ，它与第 i 个卫星的通信代价为 $h_i \cdot a^{|p-x_i|} \cdot b^{|q-y_i|}$ 。

现在你有 q 个建造通信中心的方案，你需要对每个方案，求出它与所有卫星通信代价之和（答案对 M 取模）。

【输入格式】

从文件 `satellite.in` 中读入数据。

第一行输入七个正整数 n, q, w, h, M, a, b 。

接下来 n 行，每行输入三个正整数 h_i, x_i, y_i ，表示第 i 个卫星的参数。

接下来 q 行，每行输入两个正整数 p_i, q_i ，表示第 i 个建造方案的参数。

【输出格式】

输出到文件 `satellite.out` 中。

输出 q 行，表示每种方案的答案。

【样例 1 输入】

```
4 1 9 9 1000000000 2 3
1 3 4
2 1 9
1 3 5
2 4 6
5 5
```

【样例 1 输出】

```
2620
```

【样例 1 解释】

与 4 个卫星的通信代价分别为 12, 2592, 4, 12, 总和为 2620。

【样例 2】

见选手目录下的 *satellite/satellite2.in* 与 *satellite/satellite2.ans*。

【数据范围】

对于所有数据, 保证 $1 \leq n \leq 2000$, $1 \leq q \leq 2 \times 10^5$, $10^8 \leq M \leq 10^9$, $1 \leq a, b, h_i < M$, $1 \leq x_i, p_i \leq w$, $1 \leq y_i, q_i \leq h$, $1 \leq w, h \leq 10^9$ 。

- 对于第 1 ~ 8 组数据, 保证 $n \leq 1000$, $nq \leq 100000$ 。
- 对于第 9 ~ 12 组数据, 保证 $n \leq 1000$, $w, h \leq 2000$ 。
- 对于第 13 ~ 18 组数据, 保证 $n \leq 1000$ 。
- 对于第 19 ~ 20 组数据, 没有特殊限制。

对于编号为奇数的测试点, 保证 a, b 与 M 互质。

T3 是我的你不要抢 (string)

小 ω 有很多字符串，它们都由小写字母构成。

给你 n 个字符串 a_i ，和 Q 个询问，每次对两个串 $S = a_i, T = a_j$ 询问最大的 $L(0 \leq L \leq |S|)$ 使得 $S[n-L+1 \dots |S|] = T[1 \dots L]$ 。

【输入格式】

从文件 *string.in* 中读入数据。

第一行两个正整数 n, Q ，表示一共有 n 个字符串，以及有 Q 个询问。

下面 n 行，每行一个字符串 a_i 。

下面 Q 行，每行两个正整数 x, y ，表示询问 $S = a_x$ 和 $T = a_y$ 。

【输出格式】

输出到文件 *string.out* 中。

输出 Q 行，每行一个非负整数，表示最大的 L 。

【样例 1 输入】

```
3 6
wwq
eeew
qwe
1 2
2 3
1 3
2 1
3 2
3 1
```

【样例 1 输出】

```
0
0
1
1
```

1
0

【样例 2】

见选手目录下的 *string/string2.in* 与 *string/string2.ans*。

该样例满足第二档部分分的性质。

【样例 3】

见选手目录下的 *string/string3.in* 与 *string/string3.ans*。

该样例满足第三档部分分的性质。

【样例 4】

见选手目录下的 *string/string4.in* 与 *string/string4.ans*。

该样例满足第四档部分分的性质。

【数据范围】

对于所有数据，保证 $|a_i| \geq 1, \sum_{i=1}^n |a_i| \leq 6 \times 10^5, 1 \leq Q \leq 10^6$ 。

- 对于第 1 ~ 2 组数据，保证 $n = 1$ 。
- 对于第 3 ~ 6 组数据，保证 $n \leq 10$ 。
- 对于第 7 ~ 10 组数据，保证 $\sum_{i=1}^n |a_i| \leq 5 \times 10^3$ 。
- 对于第 11 ~ 14 组数据，保证只出现 ab 两种字符，且 $\sum_{i=1}^n |a_i| \leq 5 \times 10^4$ 。
- 对于第 15 ~ 20 组数据，没有特殊限制。

T4 显然也是我整的 (graph)

出题人是懒的，数据是随的，题目是简单的。

由于小 ω 非常的巨大，所以巨大多喝水的小 ω 有一张巨大的图。

这张图有 n 个点，编号从 1 到 n ，一堆边。形式化地，我们会给你一个集合 S ，对于所有 $|i - j| \in S$ ， i 和 j 都有一条无向边。

现在小 ω 想知道这张图有多少个连通块。

当然小 ω 会问你多次，所以提醒您多测不清空，爆零两行泪。

【输入格式】

从文件 `graph.in` 中读入数据。

第一行，一个非负整数 T 。

下面 T 组数据，每组的第一行为两个正整数 n 和 m ，分别表示点数和集合大小。

每组的第二行为 m 个不同的整数 S_i ，表示集合里的元素，保证元素互不相同。

【输出格式】

输出到文件 `graph.out` 中。

输出 T 行，每一行一个整数，表示连通块个数。

【样例 1 输入】

```
3
6 2
2 3
5 1
2
658 3
219 451 581
```

【样例 1 输出】

```
1
2
```


12

【样例 2】

见选手目录下的 *graph/graph2.in* 与 *graph/graph2.ans*。

该样例满足第一档部分分的性质。

【样例 3】

见选手目录下的 *graph/graph3.in* 与 *graph/graph3.ans*。

该样例满足第三档部分分的性质。

【样例 4】

见选手目录下的 *graph/graph4.in* 与 *graph/graph4.ans*。

该样例满足第四档部分分的性质。

【数据范围】

对于所有数据，保证 $\sum m \leq 2 \times 10^5$ ， $2 \leq n \leq 10^{18}$ ， $1 \leq S_i < n$ 。

- 对于第 1 ~ 2 组数据，保证 $n \leq 10^5$ ， $\sum m \leq 50$ ， $m \leq 10$ 。
- 对于第 3 ~ 4 组数据，保证 $m \geq 1000$ ， S 集合等概率随机。
- 对于第 5 ~ 9 组数据，保证 $\sum n \leq 10^6$ 。
- 对于第 10 ~ 11 组数据，保证 $m \leq 2$ 。
- 对于第 12 ~ 14 组数据，保证 $m \leq 15$ 。
- 对于第 15 ~ 16 组数据，保证 $m \leq 30$ 。
- 对于第 17 ~ 20 组数据，没有特殊限制。