全国青少年信息学奥林匹克联赛

CCF NOIP 2023 模拟赛

时间：2023 年 11 月 1 2 日 14:00 ～ 18:30

题目名称	植物收集	美丽子区间	字符序列	网络攻防
题目类型	传统型	传统型	传统型	传统型
可执行文件名	`collect`	`interval`	`subseq`	`attack`
输入文件名	`collect.in`	`interval.in`	`subseq.in`	`attack.in`
输出文件名	`collect.out`	`interval.out`	`subseq.out`	`attack.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 MB	512 MB	512 MB	512 MB
测试点数目	10	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	`collect.cpp`	`interval.cpp`	`subseq.cpp`	`attack.cpp`

编译选项

对于 C++ 语言	`-O2 -std=c++14 -static -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/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. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 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。
5. 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。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。

植物收集（collect）

【题目描述】

Dr. Wang 是一位植物领域的专家。他要给他的学生们上一节课。课堂上需要展示一种植物。众所周知，植物的生长是有阶段的，本着严谨科学的态度，Dr. Wang 希望可以在课堂上给学生们展示该植物的每个生长阶段。

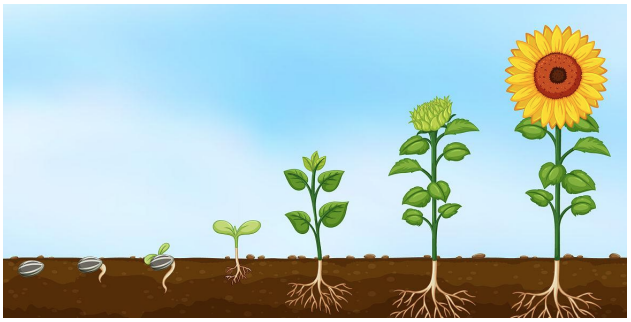

图 1: 植物“向日葵”的七个生长阶段

Dr. Wang 要讲授的植物有 n 个阶段，现在他需要弄到该植物 **每种阶段各一株** 。他打听到了这种植物每个生长阶段的价格。但由于科研经费不足，有时候直接购买并不是一个好选择。所以他计划用上他的催熟科技。具体的，Dr. Wang 可以进行如下两种操作：

- 以 a_i 的价格购买一株生长到第 i 个阶段的植物。
- 花费 k 的代价使用催熟科技，将所有已购买的植物生长阶段增加 1 。若一株植物已经到了阶段 n，则返回阶段 1 （可以理解为成熟到只剩种子，然后被重新种下去了）

现在 Dr. Wang 想让你帮忙求出，他最少需要花费多少代价，可以收集到植物的每个生长阶段。

【输入格式】

从文件 ***collect.in*** 中读入数据。

输入第一行包含两个正整数 n, k ，分别代表植物种类数和催熟科技使用一次的花费；

输入第二行包含 n 个正整数 a_i ，表示第 i 个阶段的该植物的花费。

【输出格式】

输出到文件 ***collect.out*** 中。

输出一行一个正整数表示答案。

【样例 1 输入】

```
3 10
100 1 100
```

【样例 1 输出】

```
23
```

【样例 2 输入】

```
10 5
2 9 7 1 4 7 9 3 5 1
```

【样例 2 输出】

```
28
```

【样例 3】

见选手目录下的 ***collect/collect3.in*** 与 ***collect/collect3.ans***。

【子任务】

对于 20% 的测试数据，满足 $1 \leq n \leq 6$；
对于 40% 的测试数据，满足 $1 \leq n \leq 50$；
对于 80% 的测试数据，满足 $1 \leq n \leq 2000$；
对于 100% 的测试数据，满足 $1 \leq n \leq 100000, 1 \leq k, a_i \leq 10^9$。

美丽子区间（interval）

【题目描述】

小 Z 喜欢区间。

小 Z 定义一个区间是美丽的，当且仅当这个区间的最大值或最小值不出现在区间的开头和结尾。如 $[3,2,4,1]$ 不是一个美丽的区间，因为最小值 1 出现在了结尾；而 $[2,4,1,3]$ 是一个美丽的区间，因为 $1,4$ 没有出现在开头或结尾。

小 Z 有一个排列，现在需要求出这个序列中有多少个子区间是美丽的。

【输入格式】

从文件 ***interval.in*** 中读入数据。

输入第一行一个整数 n，表示排列的长度。

第二行 n 个整数，$p_1, p_2, \ldots, p_n$，表示小 Z 的排列。

【输出格式】

输出到文件 ***interval.out*** 中。

输出一行一个整数，表示该排列中美丽的子区间的个数。

【样例 1 输入】

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

【样例 1 输出】

```
2
```

【样例 1 解释】

只有子区间 $[2,3,1,4,6,5]$ 和 $[3,1,4,6,5]$ 是美丽的。

【样例 2】

见选手目录下的 ***interval/interval2.in*** 与 ***interval/interval2.ans***。

【子任务】

对于 20% 的数据，$1 \leq n \leq 200$；
对于 40% 的数据，$1 \leq n \leq 2000$；
另有 20% 的数据，排列 p_i 随机生成；
对于 100% 的数据，$1 \leq n \leq 2 \times 10^5$.

字符序列（subseq）

【题目描述】

小 Z 喜欢字符串。

小 Z 定义了一个奇妙的函数 $f(c,s)$，对于一个长度为 n 的字符串，$f(c,s)=cs_1cs_2c\ldots cs_nc$，其中 c 是一个小写英文字母。

不难发现这个函数的作用就是在 s 的每一个空位插入一个小写英文字母，如 $f(c,\mathrm{aba})=\mathrm{cacbcac}$。

小 Z 通过这个函数构造了使用了 n 个小写字母 $c_1,c_2,\ldots,c_n$ 构造 n 个字符串 $s_1,s_2,\ldots,s_n$，其中 $s_i=f(c_i,s_{i-1})$，其中 s_0 空串。

小 Z 也非常喜欢子序列，现在他想知道 s_n 中有多少个**本质不同**的非空**子序列**。

【输入格式】

从文件 ***subseq.in*** 中读入数据。

输入第一行一个整数 n。

第二行输入一个长度为 n 的字符串，$c_1\ldots c_n$，表示小 Z 要使用的 n 个小写字母。

【输出格式】

输出到文件 ***subseq.out*** 中。

仅一行一个整数，表示 s_n 的本质不同非空子序列个数，由于个数可能很大，答案对 10^9+7 取模后输出。

【样例 1 输入】

```
2
ab
```

【样例 1 输出】

```
6
```

【样例 1 解释】

$s_n=$ bab，其本质不同子序列为 b,a,ba,ab,bb,bab。

【样例 2 输入】

```
10
abbbbaaaab
```

【样例 2 输出】

```
631522034
```

【子任务】

对于 20% 的数据，$1 \leq n \leq 4$；
对于 50% 的数据，$1 \leq n \leq 20$；
对于 100% 的数据，$1 \leq n \leq 500$.

网络攻防（attack）

【题目描述】

Dr. Wang 业务宽泛，他正在带他的学生参加网络攻防竞赛。

已知对手的网络结构可以抽象成一张 n 个节点和 m 条 **双向** 通道的连通图。各个节点之间借助通道通信。Dr. Wang 研制出了强力的干扰代码，植入后可以直接使一条通道瘫痪。但这条代码只能使用 **至多** k 次，否则会引起对手的警觉。

Dr. Wang 想要保证打击是强力的，即在打击结束之后，对方的服务器必须 **存在至少两个节点不能够继续通信** 。那么作为他的学生，你需要求出总共 **有多少种植入的方案达成目标** 。

两种方案被视为不同，当且仅当它们选择的通道数不同，或者有一条通道在一种方案被瘫痪了，而另一种方案中没有。**由于 Dr. Wang 的对手同样强大，因此干扰代码的使用次数 k 非常小，你可能需要注意其范围**

【输入格式】

从文件 ***attack.in*** 中读入数据。

第一行包含三个整数 n, m, k，含义如上。

接下来 m 行，每行包含两个数字 u, v 表示节点编号，u 到 v 之间存在双向通道。

【输出格式】

输出到文件 ***attack.out*** 中。

输出一行一个整数表示方案数。

【样例 1 输入】

```
3 3 2
1 2
2 3
3 1
```

【样例 1 输出】

```
3
```

【样例 2】

见选手目录下的 ***attack/attack2.in*** 与 ***attack/attack2.ans***。

【样例 3】

见选手目录下的 ***attack/attack3.in*** 与 ***attack/attack3.ans***。

【样例 4】

见选手目录下的 ***attack/attack4.in*** 与 ***attack/attack4.ans***。

【子任务】

对于 100% 的数据，保证 $1 \le n \le 10^5, 0 \le m \le 2 \times 10^5, k \in \{1, 2\}$，保证图是连通的。各测试点的详细数据范围如下表所示。

编号	n	m	k
1	≤ 100	≤ 200	2
2	$\le 2 \times 10^3$	$\le 2 \times 10^3$	1
3~4	$\le 2 \times 10^3$	$\le 2 \times 10^3$	2
5~6	$\le 2 \times 10^3$	$\le 2 \times 10^5$	1
7~8	$\le 2 \times 10^3$	$\le 2 \times 10^5$	2
9~12	$\le 10^4$	$\le 1.5 \times 10^4$	2
13~20	$\le 10^5$	$\le 2 \times 10^5$	2