

普及良心送分模拟赛

鸽子

May 28, 2020

题目名称	普及良	心送分模	拟赛
题目类型	传统型	传统型	传统型
目录	junior	easy	contest
可执行文件名	junior	easy	contest
输入文件名	junior.in	easy.in	contest.in
输出文件名	junior.out	easy.out	contest.out
每个测试点时限	1 秒	2 秒	4 秒
内存限制	512 MiB	512 MiB	512 MiB
子任务数目	20	10	6
是否捆绑测试	否	否	是

提交源程序文件名

对于 C++ 语言	junior.cpp	easy.cpp	contest.cpp
-----------	------------	----------	-------------

编译选项

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

注意事项

1. 不需要建立子文件夹。
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。
5. 系统栈的大小限制与内存限制相同，并且系统栈的使用计入内存使用中。

普及良 (junior)

【题目背景】

小 \$ 在 \$OI 取得了良好的成绩后, 开始研究一款奇怪的消消乐隔膜 (game)。

【题目描述】

这个隔膜会给你一个长度为正奇数的 01 字符串 S , 然后隔膜的目标是通过若干次以下操作, 使得最后的串变成一个长度为 1 的字符串 1:

1. 选择一个奇数 $i(3 \leq i \leq |S|)$,
2. 将 S 分成两个字符串 S' 和 T' , 其中 $S = S' + T'$ 且 $|S'| = i$ 。
3. 重复地将 S' 的最后三个字符组成的子串 U 替换成 $f(U)$, 直到 $|S'| = 1$ 。
4. 令新的 $S = S' + T'$ 。

其中, $f(S) = T$ 是给定的一个函数, 满足 $|S| = 3, |T| = 1$ 。

众所周知小 \$ 不会止步于简单的模拟, 所以他挖去了串的一些位置, 改成了问号 ?, 表示这位可以变成字符 0 或 1。

他想知道, 有多少种本质不同的方案, 使得把所有问号都变成字符 0 或 1 之后, 能够达成这个隔膜的目标, 答案对 $998244353 = 2^{23} \times 119 + 1$ 取模。

两种方案本质不同当且仅当存在至少一个问号在这两种方案中变成了不同的字符。

注意本题有多组测试, 注意清空数组。

【输入格式】

从文件 `junior.in` 中读入数据。

第一行输入一个正整数 T , 表示测试数据组数。

接下来 T 组测试, 在每一组中,

第一行一个长度为 8 的字符串, 表示 $f(S)$ 。其中这八位依次表示

$f(000), f(100), f(010), f(110), f(001), f(101), f(011), f(111)$

第二行一个长度为奇数的字符串, 表示 S 。

【输出格式】

输出到文件 `junior.out` 中。

对于每一组测试, 你需要输出一个范围在 $[0, 998244353)$ 的整数, 表示答案模 998244353 的值。

【样例 1 输入】

```
1
10001100
0000001
```

【样例 1 输出】

```
1
```

【样例 1 解释】

函数为: $f(000) = f(001) = f(101) = 1$, 其他的都为 0。

因为不存在问号, 所以只有 1 种填问号的方案, 即 $S = 0000001$ 。

一种可行的方案为:

1. 取出长度为 5 的前缀, 此时 $S' = 00000$, $T' = 01$ 。
2. $f(000) = 1$, S' 变为 001。
3. $f(001) = 1$, S' 变为 1。
4. 此时 $|S'| = 1$, 拼接回去, 新的 S 为 101。
5. 取出长度为 3 的前缀, 此时 $S' = 101$, $T' =$, 即空串。
6. $f(101) = 1$, S' 变为 1。
7. 此时 $|S'| = 1$, 拼接回去, 新的 S 为 1。
8. 最后 $|S| = 1$, 且 $S = 1$, 结束操作。

所以最后的答案为 1。

【样例 2 输入】

```
1
11100111
???
```

【样例 2 输出】

```
6
```

【样例 2 解释】

可行的 S 集合为 $\{000, 100, 010, 101, 011, 111\}$ 。

【样例 3】

见选手目录下的 *junior/junior3.in* 与 *junior/junior3.ans*。

【样例 4】

见选手目录下的 *junior/junior4.in* 与 *junior/junior4.ans*。

【数据范围与提示】

对于所有数据，都有 $1 \leq T \leq 10$ ， $|S| \leq 10^5$ 。

- 对于第 1 ~ 4 组数据， S 中不存在 $?$ ， $|S| \leq 2000$ 。
- 对于第 5 ~ 8 组数据， S 中不存在 $?$ 。
- 对于第 9 ~ 11 组数据， $f(S)$ 的字符串为 00010111， $|S| \leq 2000$ 。
- 对于第 12 ~ 14 组数据， $f(S)$ 的字符串为 00010111。
- 对于第 15 ~ 17 组数据， $|S| \leq 2000$ 。
- 对于第 18 ~ 20 组数据，没有额外的限制。

心送分模 (easy)

【题目背景】

众所周知，一场优秀的普及模拟赛，需要有送分题。
身为数据结构菜鸡和卡常菜鸡的小 \$，当然要出送分题了。

【题目描述】

所以，你得到了一个序列 A ，然后小 \$ 会给你 Q 次询问。
记 $f([l, r]) = \gcd(a_l, a_{l+1}, \dots, a_r)$ 。
每次询问给出区间 $[l_i, r_i]$ ，求

$$\prod_{[l', r'] \subseteq [l_i, r_i]} f([l', r'])$$

也就是，你要对询问的区间求每个子区间 \gcd 的乘积，答案对 $998244353 = 2^{23} \times 119 + 1$ 取模。

【输入格式】

从文件 *easy.in* 中读入数据。
第一行两个正整数 n, Q ，表示序列 A 的长度和询问组数。
第二行 n 个正整数 a_i ，表示序列 A 。
下面 Q 行，每行两个正整数 l_i, r_i ，表示询问的区间。保证 $1 \leq l_i \leq r_i \leq n$ 。

【输出格式】

输出到文件 *easy.out* 中。
一共 Q 行，每行一个范围在 $[0, 998244353)$ 的整数，第 i 行表示第 i 次询问的答案。

【样例 1 输入】

```
6 1
1 1 4 5 1 4
1 6
```

【样例 1 输出】

```
80
```

【样例 1 解释】

$f([3, 3]) = 4, f([4, 4]) = 5, f([6, 6]) = 4$, 其余的都为 1。

【样例 2 输入】

```
5 5
24 12 27 45 20
1 5
2 4
3 5
1 4
2 3
```

【样例 2 输出】

```
647243982
1180980
1093500
66367101
972
```

【样例 2 解释】

对于长度为 1 的区间: $f([1, 1]) = 24, f([2, 2]) = 12, f([3, 3]) = 27, f([4, 4]) = 45, f([5, 5]) = 20$ 。

对于长度为 2 的区间: $f([1, 2]) = 12, f([2, 3]) = 3, f([3, 4]) = 9, f([4, 5]) = 5$ 。

对于长度为 3 的区间: $f([1, 3]) = 3, f([2, 4]) = 3, f([3, 5]) = 1$ 。

对于长度为 4 的区间: $f([1, 4]) = 3, f([2, 5]) = 1$ 。

对于长度为 5 的区间: $f([1, 5]) = 1$ 。

对于第一组询问, 有 $\text{Ans} = 306110016000 \equiv 647243982 \pmod{998244353}$ 。

对于第二组询问, 有 $\text{Ans} = 1180980 \equiv 1180980 \pmod{998244353}$ 。

对于第三组询问, 有 $\text{Ans} = 1093500 \equiv 1093500 \pmod{998244353}$ 。

对于第四组询问, 有 $\text{Ans} = 3061100160 \equiv 66367101 \pmod{998244353}$ 。

对于第五组询问, 有 $\text{Ans} = 972 \equiv 972 \pmod{998244353}$ 。

【样例 3】

见选手目录下的 *easy/easy3.in* 与 *easy/easy3.ans*。

【样例 4】

见选手目录下的 *easy/easy4.in* 与 *easy/easy4.ans*。

【数据范围与提示】

对于所有数据，都有 $1 \leq n, Q \leq 2 \times 10^5$ ， $1 \leq a_i < 998244353$ 。

- 对于第 1 组数据， $n, Q \leq 500$ 。
- 对于第 2 组数据， $n, Q \leq 2000$ 。
- 对于第 3 ~ 4 组数据， $a_i \leq 512$ 。
- 对于第 5 ~ 6 组数据， $n, Q \leq 5 \times 10^4$ 。
- 对于第 7 ~ 8 组数据， $a_i \leq 2 \times 10^5$ 。
- 对于第 9 ~ 10 组数据，没有额外的限制。

拟赛 (contest)

【题目背景】

小 \$ 热爱构造题，所以这是一道传统题。

【题目描述】

小 \$ 给了你一个非负整数序列 A ，你需要支持单点修改。

当然这道题少不了构造，小 \$ 会多次询问你。

记 $f(x)$ 为序列 x 每个元素的异或和，例如 $f([1, 1, 4, 5, 1, 4]) = 4$ 。

每次询问会给你一个大小为 k 的可重集合 S ，你需要构造一种将 A 分割成 m 段非空序列的方案，使得分割完的序列 A'_1, A'_2, \dots, A'_m 按顺序拼接还是原来的 A ，且对于所有 $i (1 \leq i \leq m)$ ，都有 $f(A'_i) \in S$ 。

其中，你构造的方案中 m 并没有限制。

例如，对于可重集合 $\{1, 9\}$ ，序列 $[1, 9, 8, 1, 0]$ 的一种合法分割方案是 $[1], [9], [8, 1, 0]$ 。

当然，小 \$ 为了选手的体验，特地去掉了方案的输出，因此，对于每一个询问，你只要输出是否存在这样一个分割方案即可。

如果存在至少一个方案，则输出 "yes"，否则输出 "no"（不包括引号）。

【输入格式】

从文件 `contest.in` 中读入数据。

第一行两个正整数 n, Q ，表示序列长度和询问个数。

第二行 n 个非负整数 A_i ，表示序列 A 。

接下来 Q 行，每行由一个正整数 $\text{opt} \in [1, 2]$ 开头。

当 $\text{opt} = 1$ ，表示单点修改，接着一个正整数 $x (1 \leq x \leq n)$ 和一个非负整数 $y \in [0, 2^{20})$ ，表示将 A_x 修改为 y 。

当 $\text{opt} = 2$ ，表示查询，接着一个正整数 $k \in [1, 5]$ ，表示可重集合 S 的大小。接着 k 个非负整数 $s_i \in [0, 2^{20})$ ，表示可重集合里的内容。

【输出格式】

输出到文件 `contest.out` 中。

对于每一组 $\text{opt} = 2$ 的询问，即查询操作，输出一行一个字符串，表示方案是否存在。

【样例 1 输入】

```
6 6
1 1 4 5 1 4
```



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

【样例 1 输出】

```
yes
yes
yes
no
yes
```

【样例 1 解释】

对于第一组查询，可以 $[1], [1, 4], [5], [1, 4]$ 。
对于第二组查询，可以 $[1, 1, 4], [5, 1], [4]$ 。
对于第三组查询，可以 $[1, 1, 4], [5], [1, 4]$ 。
对于第四组查询，不存在方案。
经过一次修改，序列变成了 $[1, 1, 5, 5, 1, 4]$ 。
对于第五组查询，可以 $[1, 1, 5], [5], [1, 4]$ 。

【样例 2 输入】

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

【样例 2 输出】

```
yes
yes
yes
no
yes
no
```

【样例 3】

见选手目录下的 *contest/contest3.in* 与 *contest/contest3.ans*。
这组数据满足 子任务 2 的限制。

【样例 4】

见选手目录下的 *contest/contest4.in* 与 *contest/contest4.ans*。
这组数据满足 子任务 5 的限制。

【样例 5】

见选手目录下的 *contest/contest5.in* 与 *contest/contest5.ans*。
这组数据满足 子任务 6 的限制。

【数据范围与提示】

对于所有数据, 有 $1 \leq n \leq 10^5$, $1 \leq Q \leq 2 \times 10^5$ 。
对于所有数据, 有 $1 \leq k \leq 5$, $A_i \in [0, 2^{20})$ 。

【子任务】

- 子任务 1 (9 pts): 对于所有 $\text{opt} = 2$ 的询问都有 $k = 1$ 。
- 子任务 2 (16 pts): $n, Q \leq 800$ 。
- 子任务 3 (25 pts): 保证没有 $\text{opt} = 1$ 的询问。
- 子任务 4 (9 pts): 保证所有的 A_i, s_i, y 在 $[0, 2^{20})$ 里等概率随机。
- 子任务 5 (22 pts): 保证 $n, Q \leq 50000$ 。
- 子任务 6 (19 pts): 没有额外的限制。