

联考废案

Xun_Xiaoyao

2023 年 10 月 10 日 8 : 00 ~ 12 : 00

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	树状数组	子序列	新建文件夹	树题
英文题目名称	fenwick	seq	newfile	tree
可执行文件名	fenwick	seq	newfile	tree
输入文件名	fenwick.in	seq.in	newfile.in	tree.in
输出文件名	fenwick.out	seq.out	newfile.out	tree.out
每个测试点时限	2s	1s	1s	4s
内存上限	1024M	1024M	1024M	1024M
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5
结果比较方式	全文比较（过滤行末空格及文末回车）	全文比较（过滤行末空格及文末回车）	全文比较（过滤行末空格及文末回车）	全文比较（过滤行末空格及文末回车）
题目类型	传统题	传统题	传统题	传统题

二、提交源程序程序名

对于C++语言	fenwick.cpp	seq.cpp	newfile.cpp	tree.cpp
---------	-------------	---------	-------------	----------

三、优化开关

对于C++语言	-std=c++14 -O2	-std=c++14 -O2	-std=c++14 -O2	-std=c++14 -O2
---------	----------------	----------------	----------------	----------------

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. 评测时采用的机器配置为：?, 内存 ?G, 上述时限以此配置为准。
4. 感谢 moonstaring,Fyfive,Ustinian505 和 Conviction 对于这套题目的辛勤付出
5. 特别提醒：题目较为简单，AK了请不要喧哗。

一定要杜绝非知识性错误。
每题都拿分绝不是一句空话。

1 树状数组 (fenwick)

(fenwick.cpp)

1.1 题目背景

CMUST 正在学习树状数组。

1.2 题目描述

CMUST 用树状数组维护了一个长为 n 序列 $\{a_n\}$ ，初始时 a_i 均为 0。

```
1         void add(int x,int k)
2     {
3         for(;x<=n;x+=(x&-x))
4             a[x]+=k;
5     }
```

CMUST 觉得这个数据结构十分的优美，所以他想到了两种操作：

1. $1 \times k$ ，执行函数 $\text{add}(x,k)$ 。

2. $2 \mid r$ ，求 $\sum_{i=l}^r a_i \bmod 2^{64}$ 。

他希望你来解决这个问题。

注：本题输入在题目内生成。生成的方式在下发文件中给出。

你需要调用 Q 次 get_query 来获得询问，然后你需要输出，假设所有 2 操作为第 $id_1, id_2 \dots id_k$ 次询问，答案分别为 $ans_1, ans_2 \dots ans_k$ ，则你需要输出 $\bigoplus_{i=1}^k (id_i \times ans_i) \bmod 2^{64}$ 。

1.3 输入

共输入四个数 $n, Q, s1, s2$ ，分别为序列长度，询问数，和随机数参数。

1.4 输出

共一行，表示答案。

1.5 输入输出样例1

1.5.1 输入样例

5 5 1919810 114514

1.5.2 输出样例

20

1.6 输入输出样例2

1.6.1 输入样例

20 20 114514 1919810

1.6.2 输出样例

4061

1.7 输入输出样例3

1.7.1 输入样例

9129 9579 7637383388378987721 14219202285634227728

1.7.2 输出样例

1234650488382

1.8 输入输出样例4

1.8.1 输入样例

10000000 10000000 3323290814615707418 8642166001247709734

1.8.2 输出样例

5991840621884091815

1.9 样例解释

样例解释：生成的 5 个操作分别是：

1 4 4

1 2 1

1 5 3

2 3 4

1 2 3

前 3 次操作之后得到的数组是：

0, 0, 0, 4, 0; 0, 1, 0, 5, 0; 0, 1, 0, 5, 3。

所以第 4 操作查询的答案为 $0 + 5 = 5$ ，所以最终答案为 $5 \times 4 = 20$ 。

1.10 约定和数据范围

对于前 40% 的数据， $n, Q \leq 10^4$ 。对于前 70% 的数据， $n, Q \leq 10^6$ 。对于所有的数据，满足 $1 \leq n, Q \leq 10^7$ ， $0 \leq s_1, s_2 < 2^{64}$ 。

2 子序列 (seq)

(seq.cpp)

2.1 题目背景

CMUST 正在学习子序列自动机。

2.2 题目描述

给定一个序列 A ，以及他的子序列 B ，求 B 在 A 中的字典序排名，（注意，这里相同的子序列出现多次算一次）。字典序的定义（就是普通字典序的定义）：对于两个序列 A, B ， $A < B$ 当且仅当存在 i 满足 $\forall j < i, A_j = B_j \wedge A_i < B_i$ ，或者满足 $|A| < |B|$ 且 $\forall i \leq |A|, A_i = B_i$ 。

输出的排名对 998244353 取模。

2.3 输入

输入共 4 行。第一行有一个数 n ，表示 A 的长度。第二行有 n 个数，表示 A 。第三行有一个数 m ，表示 B 的长度。第四行有 m 个数，表示 B 。

2.4 输出

仅 1 行，表示答案。

2.5 输入输出样例

2.5.1 输入样例

```
4
1 2 4 1
1
2
```

2.5.2 输出样例

```
9
```

2.6 样例解释

字典序小于 B 的子序列有 $\{1\}, \{1, 1\}, \{1, 2\}, \{1, 2, 1\}, \{1, 2, 4\}, \{1, 2, 4, 1\}, \{1, 4\}, \{1, 4, 1\}$

2.7 约定和数据范围

设 A 中的最大值为 V 。

对于前 20% 的数据, $n \leq 10$ 。

对于前 40% 的数据, $n \leq 100$, $V \leq 30$ 。

对于 100% 的数据, $n \leq 10^6$, $V \leq 10^5$ 。

3 新建文件夹 (newfile)

(newfile.cpp)

3.1 题目背景

CMUST 正在学习命名。

3.2 题目描述

鉴于 CMUST 谜一般的命名方式，他最终决定起这个绝世好名字——新建文件夹！

新建文件夹里面有 n 个文件，每个文件初始为空。

现在 Larutan 觉得一堆空文件太无聊了，于是他就让 Hlet 执行了 m 操作，每次操作将区间 $[l, r]$ 内的文件都添上大小为 x byte 的后缀。

这时候 Xuxo-Y 乱入了。为了让 Larutan 身心不愉悦，他模仿 Hlet 在 Natural 的身边不停的问问题。旁边的 X-tir 听到之后，觉得问的还蛮有意思的，于是就记录了下来每一个合法的问题。

现在 X-tir 把记录下来的操作和问题分别给了 CMUST，然后他发现每一个问题形如“从文件夹的第 u 个版本到第 d 个版本中，文件夹第 l 个文件到第 r 个文件的大小总和的总和是多少”。

注：这里定义每次 Hlet 执行操作之后，文件夹进入一个新的版本。具体的，若当前正在执行第 i 个操作，则操作执行完成后，文件夹版本由 $i-1$ 变为 i 。注意，在此定义下初始文件夹是第 0 个版本。

Xuxo-Y 去找 Zaoshanghao 玩了，Larutan 去找 kxf 玩了，X-tir 去找 N-sic 玩了，Hlet 关机了，于是 CMUST 只能孤身一人解决这个问题。请你帮帮 CMUST 求出所有询问的答案吧！

由于答案可能特别大，所有询问的答案对 $10^9 + 9$ 取模。

注：先执行完所有操作，然后再回答询问。

3.3 输入

输入共 $m + q + 2$ 行。

第一行三个正整数 n, m, q ，表示文件夹有 n 个文件，Larutan 让 Hlet 执行了 m 次操作，X-tir 记录下来了 q 个问题。

第 2 到 $m + 1$ 行，每行三个整数 l, r, x ，表示一次操作。

第 $m + 2$ 到 $m + q + 1$ 行，第 i 行表示第 $i - m - 1$ 次询问，每行给出四个正整数 u, d, l, r 表示本次询问。

3.4 输出

共 q 行，第 i 行表示第 i 次询问的答案。

3.5 输入输出样例1

3.5.1 输入样例

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

3.5.2 输出样例

```
169
39
99
75
72
9
67
129
164
204
```


3.6 输入输出样例2

3.6.1 输入样例

```
10 10 10
1 10 20
3 9 85
1 8 4
2 8 57
1 10 16
1 7 29
1 10 50
3 9 76
2 9 75
2 8 45
4 6 5 9
10 10 5 8
3 6 5 9
0 8 2 6
8 9 5 7
4 7 3 6
3 5 3 10
5 6 5 10
4 5 5 6
5 8 4 7
```

3.6.2 输出样例

```
2554
1799
3095
6284
2247
3280
3149
1857
696
3964
```

3.7 约定和数据范围

对于 100% 的数据，保证 $1 \leq l, r \leq n \leq 10^5$ ， $0 \leq u, d \leq m \leq 10^5$ ， $q \leq 10^5$ 。

测试点编号	$n, m, q \leq$	$x \leq$
1 ~ 2	50	1
3 ~ 4	50	100
5 ~ 8	5000	1
9 ~ 10	5000	10^9
11 ~ 14	10^5	1
15 ~ 20	10^5	10^9

4 树题 (tree)

(tree.cpp)

4.1 题目背景

CMUST 正在学习板子套板子

4.2 题目描述

给定根为 1 的树 T_1, T_2 , T_1 的树边 $e(u, v)$ 有边权 $w(u, v)$ 。两棵树的点集相同, 均为 $1 \sim n$, 编号为 i 的点有权值 A_i 。定义两点的路径长度 $dis(u, v)$ 为树 T_1 上 u, v 之间简单路径上的边的权值和; 定义路径 $P(u \rightarrow v)$ 为树 T_2 上 u, v 之间简单路径上的点对应的有序集合, $P(u \rightarrow v)_i$ 表示路径上的第 i 个点; 定义 sub_u 为树 T_2 上 u 子树内的点的集合 (包含 u)。

你需要对于所有 u 求出 $ans_u = \max_{v \in sub_u} dis(u, v) + \sum A_{P(v \rightarrow u)_i} \times (i - 1)$ 。

4.3 输入

输入共 5 行。

第 1 行一个正整数 n , 表示点数。

第 2 行 $n - 1$ 个正整数, 第 $i - 1$ 个数 f_i 表示 T_1 上编号为 i 的点的父亲, 保证 $f_i < i$ 。

第 3 行 $n - 1$ 个整数, 第 $i - 1$ 个数 W_i 表示 T_1 上 $w(i, f_i)$ 。

第 4 行 $n - 1$ 个正整数, 第 $i - 1$ 个数 g_i 表示 T_2 上编号为 i 的点的父亲, 保证 $g_i < i$ 。

第 5 行 n 个整数, 第 i 个数表示 A_i 。

4.4 输出

输出一行共 n 个数, 第 i 个数表示 ans_i 。

4.5 输入输出样例

4.5.1 输入样例

```
5
1 2 3 1
9 5 6 9
1 1 1 3
-8 5 -6 5 -7
```

4.5.2 输出样例

```
12 0 17 0 0
```

4.6 约定和数据范围

对于所有测试数据： $1 \leq n \leq 5 \times 10^5$, $-10^9 \leq W_i \leq 10^9$, $-10^6 \leq A_i \leq 10^6$ 。

测试点编号	n	$T1$	$T2$
1 ~ 2	≤ 10	无	无
3 ~ 4	≤ 500	无	无
5 ~ 6	≤ 5000	无	无
7 ~ 8	$\leq 5 \times 10^5$	$f_i = i - 1$	$g_i = i - 1$
9 ~ 10	$\leq 5 \times 10^5$	$f_i = i - 1$	无
11 ~ 12	$\leq 5 \times 10^5$	树形态随机	树形态随机
13 ~ 14	$\leq 5 \times 10^5$	无	树形态随机
15 ~ 20	$\leq 5 \times 10^5$	无	无