

# ВШЭ АиСД 2021. Списки (noSTL)

4 дек 2021, 18:54:39

старт: 8 окт 2021, 12:00:00

финиш: 17 окт 2021, 23:59:59

длительность: 9д. 11ч.

начало: 8 окт 2021, 12:00:00

конец: 17 окт 2021, 23:59:59

## Е. Всё за корень\* (0.4)

Ограничение времени	1.25 секунд
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод
Вывод	стандартный вывод

После выставления результатов очередного контеста по ~~лучшему предмету на втором курсе~~ АиСД оказалось, что несколько человек были перепутаны, у кого-то была проверена не последняя версия, а кого-то вообще пропустили. Теперь приходится всех допроверять и перепроверять. Никитка очень устал, помогите ему проавтоматизировать проверку.

Напишите структуру данных, которая эффективно будет обрабатывать запросы следующего вида:

0. Построение структуры из массива  $a$  (здесь конечно не корень).

1. `insert(i, x)` - вставка  $x$  в  $i$ -ую позицию.

2. `erase(i)` - удаление элемента на  $i$ -ой позиции.

3. `change(i, x)` - замена элемента по индексу  $i$  на  $x$ .

4. `at(i)` возвращает элемент по индексу  $i$ .

5. `compute(l, r)` возвращает результат операции (определённой в формате ввода) на отрезке  $[l, r]$  структуры.

6. `size()` возвращает размер структуры.

## Формат ввода

В первой строке вводится один из двух видов групповой операции:

0. `+` `<mod>` - сложение по модулю  $mod \in [1; 10^9]$

1. `^` - побитовый xor

На следующей строке даётся целое число  $n \in [1; 2 \cdot 10^6]$  - размер исходного массива.

На следующей строке через пробел поступает целочисленный массив (элементы которого лежат в диапазоне  $[-10^9; 10^9]$ ), который нужно занести в структуру.

Затем вводится целое число  $q \in [1; 2 \cdot 10^3]$  - кол-во запросов. На следующих  $q$  строках вводятся запросы в следующем виде:

1. `insert <i> <x>`. Вставка  $x$  в  $i$ -ую позицию.

Гарантируется, что  $i \in [1; \text{size()} + 1]$ ,  $x \in [-10^9; 10^9]$ .

2. `erase <i>`. Удаление элемента на  $i$ -ой позиции.

Гарантируется, что  $i \in [1; \text{size()}]$ .

3. `change <i> <x>`. Замена элемента по индексу  $i$  на  $x$ .

Гарантируется, что  $i \in [1; \text{size()}]$ ,  $x \in [-10^9; 10^9]$ .

4. `at <i>`. Возвращает элемент по индексу  $i$ .

Гарантируется, что  $i \in [1; \text{size()}]$ .

5. `compute <l> <r>`. Возвращает результат операции на отрезке  $[l, r]$  структуры.

Гарантируется, что  $1 \leq l \leq r \leq \text{size()}.$

6. `size`. Возвращает размер структуры.

## Формат вывода

На все запросы каждого из типов at, compute, size в отдельной строке выведите ответ.

Пример 1

Ввод	Вывод
+ 11	10
4	100
1 2 3 4	7
4	
compute 1 4	
insert 1 100	
at 1	
compute 1 4	

Пример 2

Ввод	Вывод
^	4
4	3
1 2 3 4	3
4	
compute 1 4	
insert 1 3	
at 1	
compute 1 4	

Примечания

1. Эталонное решение не использует никакие алгоритмы / структуры данных, которые вам не давались, лишь идеи с лекций и семинаров.
2. Использование декартова дерева запрещено.
3. Кому нечем заняться: попробуйте реализовать еще вычисление умножения по простому модулю. (не оценивается; там еще в сложности дополнительный множитель появится, поэтому в TL на макс тесте не пройдет)

by Игумнов Никита

Система оценки

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n$	$q$		
0	1	—	—	—	Тесты из условия.
1	0,5	$n \leq 5$	$q \leq 5$	0	
2	1,5	$n \leq 2 \cdot 10^3$	$q \leq 100$	0 – 1	
3	3	$n \leq 2 \cdot 10^5$	$q \leq 10^3$	0 – 2	
4	1	$n \leq 6 \cdot 10^5$	$q \leq 10^3$	0 – 3	Offline-проверка
5	1	$n \leq 2 \cdot 10^6$	$q \leq 2 \cdot 10^3$	0 – 4	Offline-проверка

Набрать здесь

Отправить файл

1

Отправить

Предыдущая