

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

Дисциплина «Методы программирования»

Лабораторная работа №6

на тему «Деревья»

Выполнила студентка группы ВКБ32

Куринский К.И.

г. Ростов-на-Дону

2021 г.

Задание:

Реализуйте структуру данных для эффективного вычисления номера максимального из нескольких подряд идущих элементов массива.

Входные данные:

В первой строке вводится одно натуральное число N (1 ≤ N ≤ 100000) — количество чисел в массиве.

Во второй строке вводятся N чисел от 1 до 100000 — элементы массива.

В третьей строке вводится одно натуральное число K (1 ≤ K ≤ 30000) — количество запросов на вычисление максимума.

В следующих K строках вводится по два числа — номера левого и правого элементов отрезка массива (считается, что элементы массива нумеруются с единицы).

Выходные данные:

Для каждого запроса выведите индекс максимального элемента на указанном отрезке массива. Если максимальных элементов несколько, выведите любой их них.

Числа выводите в одну строку через пробел.

Листинг задания:

#include <bits/stdc++.h>

#define int int64\_t

#define t feuer\_frei

using namespace std;

int a[100000];

struct segtree

{

int MAXN = 1000000;

int t[400000];

int n;

int sz = -1;

void build(int v, int l, int r)

{

if (l == r)

{

t[v] = a[l];

}

else

{

int mid = (l + r) / 2;

build(v \* 2, l, mid);

build(v \* 2 + 1, mid + 1, r);

t[v] = t[v \* 2] + t[v \* 2 + 1];

}

}

int get(int v, int l, int r, int ql, int qr)

{

if(l > r){

return 0;

}

if (l > qr || r < ql)

{

return 0;

}

if (ql <= l && r <= qr)

{

return t[v];

}

int mid = (l + r) / 2;

return get(v \* 2, l, mid, ql, qr) + get(v \* 2 + 1, mid + 1, r, ql, qr);

}

void upd(int v, int l, int r, int pos, int val)

{

if (l == r)

{

t[v] = val;

a[l] = val;

}

else

{

int mid = (l + r) / 2;

if (pos <= mid)

{

upd(v \* 2, l, mid, pos, val);

}

else

{

upd(v \* 2 + 1, mid + 1, r, pos, val);

}

t[v] = t[v \* 2] + t[v \* 2 + 1];

}

}

};

int32\_t main()

{

int n, m;

cin >> n;

for(int i = 0; i < n; i++)

{

cin >> a[i];

}

segtree t;

t.build(1, 0, n - 1);

cin >> m;

while(m--)

{

int l, r;

char c;

cin >> c;

cin >> l >> r;

if(c == 's')cout << t.get(1, 0, n - 1, l - 1, r - 1) << endl;

if(c == 'u')t.upd(1, 0, n - 1, l - 1, r);

}

}

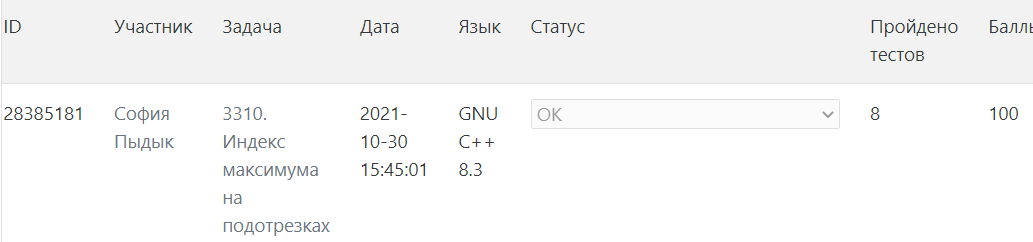


Рисунок 1 – результат работы задания.