

Национална школа по информатика
Пазарджик, 21 – 27 август 2018 година
9- 10 клас

Задача В3. ЗАЯВКИ ЗА СУМА

Ави си играе с множеството от числа A . Първоначално то е празно, но момичето може да добавя числа в него и да премахва вече добавени числа. Никога обаче Ави не добавя число, което вече е в A , нито се опитва да изтрие такова, което не се съдържа в множеството.

Индекс на дадено число от A ще наричаме броя на числата в A , по-малки от него, т.е. броя числа, които биха били преди даденото, ако числата от A се подредят във възходящ ред. С a_i обозначаваме числото от A с индекс i . Ави се интересува какво би се получило ако вземе няколко числа с последователни индекси $(a_l, a_{l+1}, \dots, a_r)$, и изчисли $l \cdot a_l + 2 \cdot a_{l+1} + \dots + (r-l+1) \cdot a_r$. Тъй като полученият отговор може да бъде много голям, Ави иска само остатъка му при деление на $10^9 + 7$.

Вашата задача е да напишете програма, която:

- Започва с множеството A , което първоначално е празно.
- Поддържа добавяне и премахане на числа.
- Отговаря на въпросите на Ави от горния вид.

Детайли по имплементацията

Вие трябва да предадете към системата програмен файл **sumquery.cpp**, който ще се компилира заедно с програма на журито и в който трябва да са реализирани описаните по-долу функции. Вашата програма трябва да съдържа в началото си `#include "sumquery.h"`.

Вашата програма трябва да реализира следните функции:

- `void init(int q)`
 - Тази функция се извиква веднъж за всяко изпълнение на програмата, преди извикването на останалите функции.
 - q : Общият брой на извикванията на останалите функции.
- `void add_number(long long x)`
 - Извикването на тази функция обозначава *добавяне* на числото x в A .
 - x : добавяното число в множеството. Гарантира се, че $0 \leq x \leq 10^{18}$ и че x не се съдържа в A .
- `void remove_number(long long x)`
 - Извикването на тази функция обозначава *премахане* на числото x от A .
 - x : премаханото число от множеството. Гарантира се, че x се съдържа в A .
- `int sum_query(int l, int r)`
 - Функцията трябва да връща остатъка на $l \cdot a_l + 2 \cdot a_{l+1} + \dots + (r-l+1) \cdot a_r$ при деление на $10^9 + 7$
 - l и r : първият и последният индекс на последователността от числа, за която се отнася въпросът. Гарантира се, че $0 \leq l \leq r < \langle \text{брой числа в } A \rangle$

Освен горните функции Вашата програма може да ползва и реализира други вътрешни функции и глобални променливи. Тя трябва да *не съдържа* функция `main`.

В системата ще намерите прикачени файловете `sumquery.h`, `Lgrader.cpp` (грейдър, с който можете да тествате на локалния си компютър), `sumquery.cpp` (скелет на програмния файл, който трябва да предадете) и входен и изходен файл за примера от условието.

Национална школа по информатика
Пазарджик, 21 – 27 август 2018 година
9- 10 клас

Пример

Извикана функция	Множеството A	Очакван отговор
init(7)	{}	N/A
add_number(4)	{4}	N/A
add_number(2)	{2, 4}	N/A
add_number(3)	{2, 3, 4}	N/A
sum_query(0, 1)	{2, 3, 4}	$8 = (1*2) + (2*3)$
sum_query(1, 2)	{2, 3, 4}	$11 = (1*3) + (2*4)$
remove_number(3)	{2, 4}	N/A
sum_query(1, 1)	{2, 4}	$4 = (1 * 4)$

Подзадачи

Нека с Q обозначим общия брой на извикванията на `add_number`, `remove_number` и `sum_query`. Тестовите са групирани в подзадачи, като точките за дадена подзадача се получават само ако програмата Ви премине всички тестове успешно:

1. (10 точки) $Q \leq 1000$
2. (40 точки) $Q \leq 200\,000$, $l=r$ за всяко извикване на `sum_query`
3. (50 точки) $Q \leq 200\,000$

За всички подзадачи `add_number` и `remove_number` се викат с $0 \leq x \leq 10^{18}$

Локално тестване

За да можете да тествате решението си на компютъра си, Ви се предоставят файловете `Lgrader.cpp` и `sumquery.h`, които да компилирате заедно с Вашето решение `sumquery.cpp`.

Вход

- Ред 1: едно цяло число Q .
- Редове 2 до $(Q + 1)$
 - 1 x : извиква се `add_number(x)`
 - 2 x : извиква се `remove_number(x)`
 - 3 $l r$: извиква се `sum_query(l, r)`

Изход

На отделни редове ще се изведат резултатите от всяко извикване на `sum_query`.