

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА БАЗИ ДАННИ

Първото решение, което ще разгледаме, е базирано на идеята за разделяне на операциите на блокове. Нека обработваме по K операции едновременно. С линейно обхождане може да намерим новите стойности на числата в редицата след изпълнението на въпросните заявки. Основната идея е да разделим новодобавените стойности по x и по y . Стойността x , която добавяме към числата, се увеличава в началото на интервала на всяка операция и намалява със същата стойност в края му. Същото се случва и със стъпката, с която се увеличава y стойността на всяко следващо число. Важно е да се отбележи, че y стойността намалява внезапно след края на интервала на операцията с $u_i * (r_i - l_i)$. Това намаляване е по-удачно да се симулира чрез изваждане от x стойността.

След като сме намерили новите стойности на числата, може да разгледаме кои от тях са достигнали желаната стойност по време на изпълнението на последните K заявки. Тъй като за всяко число това се случва най-много веднъж, може да изпълним всяка от операциите върху конкретното число, за да открием тази, която ни трябва. Сложността на решението е $O\left(\frac{Q}{K} * N + K * N\right)$ и затова е удачно да се избере стойност за K , близка до \sqrt{N} .

Второто решение използва структурата от данни персистентно сегментно дърво. Чрез него можем да изпълняваме операции за добавяне на аритметична прогресия върху интервал от дървото. Това се постига с помощта на две *lazy* стойности съответно за x и y . Тъй като дървото е персистентно, след изпълнението на всички заявки можем да възстановим стойността, която някое число е получило след всяка от операциите. Така с двоично търсене за всяко число може да се намери номерът на търсената операция. Макар асимптотично това решение да има сложност $O(N * \log_2^2 N)$ и да е теоретично по-бързо от първото, на практика двете работят почти еднакво бързо.

Изготвил анализа:
Добрин Башев