

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА РЕДИЦА

Задачата се решава с т.нар. “greedy” (лаком) алгоритъм. Първо сортираме елементите на редицата във възходящ ред. Важно наблюдение е, че никога нямаме полза да избираме елемент, който е най-малък в редицата (и съответно да увеличим всички останали с 1). Започваме да разглеждаме редицата от най-малките елементи към най-големите, като ги разглеждаме по „групи“ от равни елементи. Нека след сортирането се е получило, например, „1 1 1 3 3 5 7 7“. Както казахме, няма никакъв смисъл да избираме за елемент на операцията някоя от единиците. Затова и първата група която разглеждаме е тази от тройките. Целта на всяка стъпка на алгоритъма е да оеднакви всички елементи от началото на редицата до групата, която разглеждаме (включително). Т.е., първата стъпка на нашия алгоритъм ще се опита да оеднакви групата „3 3“ с групата „1 1 1“. Целта е след тази стъпка на алгоритъма да имаме пет равни по стойност елемента. Това става (с минимална цена), като изберем всеки елемент от групата, която разглеждаме в момента (3 3) толкова пъти, колкото е разликата по стойност между елементите в групата и елементите преди групата. Важно е да се отбележи, че разглеждайки последователно (в нарастващ) ред групите от еднакви елементи и изпълнявайки гореописаната стъпка, гарантираме, че елементите от началото на редицата до групата, която разглеждаме, ще бъдат с еднаква стойност. Нека се върнем на групата „3 3“. Изпълнявайки описаната стъпка (избирайки всеки елемент от групата по толкова пъти, колкото е разликата с елементите от началото на редицата до групата, която е $3 - 1 = 2$ в случая) ще получим редица „5 5 5 5 9 11 11“. Ново важно наблюдение е, че разликата между групата, която наблюдавахме в тази стъпка и следващата по големина група е нарастнала със стойността на разликата, която беше между групата, която разглеждахме в тази стъпка и елементите от началото на редицата. Разликите между останалите групи се запазват същите.

Стойността, която трябва да добавяме като цена на всяка стъпка се изразява с:

$$((a[i] - a[i-1]) + prev) * count$$

Където $a[i]-a[i-1]$ е разликата между стойността на елементите в текущо разглежданата група и елементите от началото на редицата, $prev$ е стойността на същата разлика от предната стъпка (като на първа стъпка тя, логично, е 0), а $count$ е броят на елементите в текущо разглежданата стъпка.

Сложността на описания алгоритъм е $O(N * \log N)$ заради сортирането на елементите, самият алчен алгоритъм е с линейна сложност.

Друг алчен алгоритъм, който също ще даде минималната цена, която е нужна да оеднакви елементите в редицата, е на всяка стъпка да избираме текщо най-големият елемент в редицата. Ако реализираме този алгоритъм и симулираме всяка стъпка (т.е. да увеличаваме всички елементи без (един от) максималния) означава, че трябва

направим толкова стъпки, колкото е отговора на задачата и на всяка такава стъпка с $O(N)$ трябва да намерим най-големия елемент и отново с $O(N)$ да увеличим всички останали елементи с единица. Решение с такава сложност ще получи 40 точки.

Пълно изчерпване на всички възможни ситуации (с важното съображение, че избор на минималния елемент в редицата със сигурност не води до оптимално решение) ще получи 20 точки.

Автор: Момчил Иванов