

## АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА

### ИЗПИТИ

Нека в масивът `max[]` сме запазили максималните точки за всеки един изпит, а в масивът `current[]` сме запазили текущият резултат.

Решението за 50 точки е да проверяваме за различните брой дни, започвайки от 0 и увеличавайки ги с 1, кога точно ще получим нужния сбор  $M$ , като на всяка итерация трябва да пресметнем колко е текущият бал. Това правим по следния начин: разглеждаме в момента `days` дни за учене, тогава  $currentSum = \text{Min}(\text{max}[i], \text{current}[i] + \text{days})$ ,  $i = 0..n-1$ . Т.е. за всеки изпит  $i$  взимаме или максимумът който можем да получим за този изпит, в случай, че за `days` дни можем да постигнем този максимум, или  $current[i] + \text{days}$ , в случай, че за `days` дни не можем да постигнем този максимум. При първата такава сума по-голяма или равна на  $M$ , сме намерили решението на задачата.

Решението за 100 е аналогично на това за 50, но вместо да започнем с 0 `days` и постепенно да увеличаваме дните, с които пробваме да постигнем сумата, то използваме техниката "двоично търсене по отговора". Т.е., имаме интервал, в който търсим отговора на задачата, като `left` очевидно е 0 (т.е. още с текущите знания е възможно да покрием търсеня бал), а максималния брой дни (или `right` на този интервал) е максималната разлика  $\text{max}[i] - \text{current}[i]$ , защото колкото и повече дни от тази максимална разлика да учим, то ние ще сме постигнали максимума като точки за всеки един изпит и няма да има смисъл да продължаваме с ученето. И така, при установени начални `left` и `right`, правим класическо двоично търсене. На всяка итерация гледаме  $\text{mid} = (\text{left} + \text{right}) / 2$  като "проба" за отговор на задачата. Нека  $currentSum = \text{балът}$ , който получаваме при учене `mid` дни. Ако  $currentSum < M$ , то ще са ни нужни със сигурност повече от `mid` дни учене за постигане на бал  $M$  и продължаваме двоичното търсене с  $\text{left} = \text{mid} + 1$ ,  $\text{right} = \text{right}$ . Ако ли пък  $currentSum \geq M$ , то отговорът на задачата е в интервала  $\text{left} = \text{left}$ ,  $\text{right} = \text{mid}$ . Останалото е въпрос на имплементационна техника да открием "дъното" на двоичното търсене и да върнем правилния отговор. Така получаваме решение със сложност  $O(N * \log \text{MAX\_RESULT})$ , където `MAX_RESULT` е максималният възможен брой дни, който би ни се наложило да учим.

Съществува и решение със сложност  $O(N * \log N)$ , което е необходимо за сортирането на данните първоначално, след което самото решение на задачата се намира с  $O(N)$ , но него оставяме за упражнение на читателя. Авторът на задачата имаше намерение това решение да „взема“ 100, а това с двоично търсене по отговора на „вземе“ само 80 точки, но разграничаването на тези две решения се оказа не лека задача. *Подсказка: сортирането на данните се прави по разликата  $\text{max}[i] - \text{current}[i]$  в нарастващ ред.*

Автор: Момчил Иванов