

## Задача 2. Грейдър - Решение

---

Автор: Слави Маринов

Да разгледаме най-простите случаи за входните данни на тази задача:

- Имаме само една или две задачи за тестване

Очевидно просто ще съберем времената за обработка на задачите и това ще ни даде решението

- Имаме три задачи за тестване

Тук имаме два избора: или тестваме всяка задача поотделно, или ги комбинираме трите заедно. Решението ни е по-добрият от тези два варианта.

- Имаме четири задачи за тестване

Тук пак имаме два избора: Да тестваме отделно последната задача, след което остават три задачи (тях вече знаем как да ги обработим), или да комбинираме последните 3 задачи и да ги тестваме заедно, и да тестваме отделно първата задача. Отново избираме по-бързия от двата варианта

- Имаме пет задачи за тестване

Отново имаме два избора: Тестваме отделно последната задача, след което остават четири задачи (тях знаем как да ги обработим), или да комбинираме последните 3 задачи и да ги тестваме заедно, и остават първите две задачи (за които отново знаем как да ги решим). Избираме по-добрият от двата варианта.

Забелязвате ли схемата за решаване? В общи линии, всеки път избираме между два варианта: да тестваме отделно последната задача или да я тестваме заедно с предните две. За Остатъка, понеже съдържа по-малко задачи, вече знаем отговора.

Или казано на малко по-компютърен език:

Минимумът за първите  $K$  задачи = по-малкото-от ( времето за задача  $K$  + минимумът за първите  $(K - 1)$  задачи, времето за задачи  $(K, K-1, K-2)$  взети заедно) + минимумът за първите  $(K - 3)$  задачи)

Или, още по-компютърно:

```
solution[k] = min(solution[k - 1] + t[k], solution[k - 3] + group_t[k - 2]),
```

където:

$t[k]$  е времето за обработка на задачата с номер  $k$  сама,  
 $group\_t[k - 2]$  е времето за обработка на задачи с номера  $k - 2, k - 1, k$  заедно.

Така, изчислявайки първо  $solution[1]$ , после  $solution[2]$ ,  $solution[3]$  и т.н., в  $solution[n]$  ще получим отговора.

Сложността на това решение е  $O(N)$ .