

Напаст (анализ)

A2 / 100



Задача „Напаст“ се решава чрез метода на динамичното оптимиране (dynamic programming).

Сложността на алгоритъма който я решава е $O(N \cdot p)$, като p представлява броя на простите числа по-малки от k .

В началото на изпълнението на програмата пресмятаме простите числа които са по-малки от k и ги запазваме във вектор (условно наречен `primes`).

След това създаваме масив (условно наречен `dp`) в който във клетката с индекс i записваме броя начини по който i може да се представи като сума на прости числа по-малки от k .

Броя начини по които можем да представим m като сума на прости числа ще намираме чрез формулата.

$$\sum_{i=0}^{n.size()-1} dp[m - n[i]]$$

Като съответно в началото трябва да зададем 0 на всички стойности които не могат да се получат като сума на прости числа по-малки от k . Такива със сигурност са 1,2,3,4, без значение от стойността на k . Също така трябва да внимаваме да не излезем от границите на масива `dp`, като това става с елементарни проверки.

Важно е да отбележим, че в задачата се изисква да изведем отговора под модул, тъй като търсените стойности могат да бъдат крайно големи и неудобни за съхранение в стандартните типове. За това просто при извършване на всяко елементарно пресмятане по време на попълването на масива `dp` трябва да вземаме резултата под модул. Това няма да доведе до промяна в крайния отговор или допълнителни пресмятания, тъй като $(a+b) \bmod c = (a \bmod c + b \bmod c) \bmod c$.

В края на изпълнението на програмата просто трябва да изведем стойността на `dp[m]`.