**Анализ**

Решение за 20 точки: Понеже ***N*** e малко тук може да се направи пълно изчерпване – генерират се всички възможни редици с дължина ***N*** и елементи от 0 до ***K***-1. За всяка се проверява дали е *интересна*. Очакваната сложност е .

Решение за 50 точки: Това е първата стъпка към цялостната идея – ще подходим с динамично оптимиране. Нека имаме двумерно динамично *dp[len][last]* – броят *интересни* редици с дължина*len*и последно число *last*. Лесно се вижда, че зависимостта е следната: за (започнали сме вече проста редица) и (започваме нова проста редица, а в предния елемент сме завършили някоя друга). Базовите ни стойности са , а за . Нашият отговор всъщност ще представлява сумата: . Имаме двумерно динамично, което се запълва със сложност .

Решение за 100 точки: Важно е да забележим, че всъщност и едномерно динамично ще ни свърши работа. Нека *dp[len]* е броят *интересни* редици с дължина*len*. Тук зависимостта се намира по следния начин – разглеждаме всички възможни случаи за дължината на последната проста редица. Ако *len* е достатъчно голямо, очевидно последната проста редица може да е с дължина 1, 2, … ***K***. Съответно за всяка от тези стойности броят интересни редица са , *, …* . А *dp[len]* е равно на тази сума. Ако смятаме сумата всеки път няма да подобрим времевата сложност. Затова ще оптимизираме вътрешния цикъл – в една променлива ще пазим сумата на последните ***K*** динамични. Сравнително лесно я поддържаме – като намерим *dp[len]*, прибавяме неговата стойност към сумата и изваждаме . Има и алтернативен подход с префиксни суми. Окончателната сложност е .

*Автор: Илиян Йорданов*