# Упражнения: Едномерно динамично оптимиране

## Сума 1-3-5

Да се напише програма, която казва по колко начина можем да представим сумата N използвайки числата 1, 3 и 5.

### Пример

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 6 | 8 |

### Решение

Нека да фиксираме базовите случаи:

Ако **N < 0**, очевидно няма как да намерим решение, понеже няма как да се образува отрицателна сума от зададените числа (1, 3 и 5). Тогава броят на начините е 0.

Ако **N = 0** ще връщаме стойност 1, понеже има точно един начин да получим сума 0 и това е да не вземаме **нито едно** от числата (някой някъде в условието казал ли е, че трябва непременно числата да участват ☺???)

Ако **N > 0** ще изчисляваме отговора по следния начин:

Решението се получава като съберем начините:

* по които можем да получим решение, ако на последната стъпката сме използвали 1 за сумирането
* по които можем да получим решение, ако на последната стъпката сме използвали 3 за сумирането
* по които можем да получим решение, ако на последната стъпката сме използвали 5 за сумирането

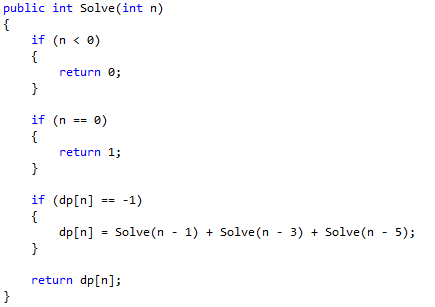
Това може да се изрази рекурсивно:

**Solution(N) = Solution (N-1) + Solution(N-3) + Solution(N-5)**

Тук разглеждаме подзадачите със състотяния изразени като **N - стойност**, понеже така се отбелязва от къде сме „дошли“. Иначе казано, ако за да стигнем до стъпка N сме добавили 1 към сумата, то значи сме „дошли“ от състояние **N-1**, и т.н.

С цел да се напише ефективна програма, резултатите от рекусивните извиквания могат да се пазят в масив. Така при повторно извикване няма да се смята от съответната стойност до базовите случаи, а директно ще се връща вече изчисления отговор. Този похват се нарича **меморизация (memorization)** и е широко използван в динамичното оптимиране. С негова помощ може и лесно да се възстанови начин, за който се получава съответния резултат.

Една примерна реализация на функцията за решаване на задачата би била следната:



Важно е да имате предвид, че в началото масива **dp** трябва да бъде запълнен от **-1**, за да познаваме лесно къде реално нямаме изчислено решение. Често стойността **-1** се използва в динамичното оптимиране за отбелязване на липса на решение или неизчислено към момента такова.