

Задача "Лесенка"

C++. Динамическое программирование. Примеры решения з...



Задача "Лесенка"

На лестнице, состоящей из n ступеней, записаны числа a_1, a_2, \dots, a_n .

Необходимо пройти по лестнице, стартуя перед первой ступенькой (можно считать, что старт начинается на нулевой ступени, которая соответствует поверхности земли), и дойти до последней ступени. На каждом шагу можно перейти на следующую по счёту ступень или перешагнуть через одну ступень. Необходимо найти максимальное значение, которое может принимать сумма чисел, записанных на тех ступеньках, на которые мы наступили при прохождении лестницы.

Пусть $dp[i]$ — это максимальное значение суммы чисел на пройденных ступеньках, если мы шли и остановились на ступени номер i . Для заполнения массива dp будем использовать восходящий способ.

Для удобства заполнения введём так называемый барьерный элемент, а именно заполним значение $dp[0] = 0$. Далее заполним значение $dp[1] = a[1]$. Для всех ступеней, начиная со второй, будем находить значение по следующей формуле перехода: $dp[i] = \max(dp[i - 1], dp[i - 2]) + a[i]$, потому что на ступень i мы могли попасть или непосредственно со ступени $i - 1$, или $i - 2$. Нам в любом случае необходимо заранее заполнить два стартовых значения в

массиве dp , а барьерный элемент упрощает заполнение, так как пару значений $dp[0]$ и $dp[1]$ заполнить вручную проще, чем пару значений $dp[1]$ и $dp[2]$.

При решении этой задачи хорошо видно, как мы используем свойство оптимальной подструктуры. Чтобы найти оптимальную стоимость, за которую можно дойти до ступени i , мы рассматриваем ступени, из которых мы в неё могли прийти, и используем оптимальную стоимость, за которую можно дойти до этих ступеней, чтобы посчитать ответ для ступени i .

В итоге ответ будет равен значению $dp[n]$.