



Вау! ИИ готовит к ЕГЭ по информатике

Попробовать

Урок Спринт 1.7

Задача о сдаче

Постановка задачи

Обратимся к еще одному примеру. После продажи части содержимого рюкзака, мы пойдём в магазин с крупными купюрами денег. К нашему несчастью, в кассе остались только монеты, поэтому сдача с наших покупок будет мешком с монетами. Нам хотелось бы знать, сколько существует способов получения сдачи с сохранением номинала. Для начала разберём небольшой пример, когда у кассира много монет номиналом 2, 3 и 5 рублей (будем считать, что количество монет неограниченно), а сдачу мы ожидаем в размере 12 рублей. Очевидно, что оптимальным для нас будут две монеты по 5 рублей и одна — 2 рубля. Есть ли другие варианты? Да, есть, но монет в них будет больше, например 6 по 2 рубля.

Решение

Теперь попробуем составить алгоритм и написать программу. Итак, пусть $N = 12$ — размер сдачи, а слайс `coins` содержит номиналы монет. Нам нужно придумать, как, используя набор монет, определить количество способов получения 12 рублей.

Основываясь на понятии динамического программирования, нам нужно выяснить, что нам нужно добавить к предыдущим решениям вместо перерасчета тех же значений.

Очевидно, что нам придется перебирать весь слайс монет. Нам также нужно проверять, больше ли монета значения N . Кроме этого, нам понадобится слайс, который будет содержать значения возможных способов получения сдачи до значения N . Индексами данного слайса будут номиналы сдачи, которую мы хотим получить. Рассмотрим пример:

```
coins := []int{2,3,5} // доступные монеты
ways := make([]int, N + 1) // слайс с возможными вариантами
ways[0] = 1 // сдачу размером ноль можно получить только одним способом
```

Если мы начнем перебирать весь слайс монет и сравнивать элементы со слайсом способов, то мы определим, сколько раз монету можно использовать для получения значений в слайсе способов:

```
for _, coin := range coins {
    // переберем все значения слайса с количеством способов
    for j := range ways {
        // если монета меньше значения, которое нам нужно получить
        if coin <= j {
            // обновим значение с учетом предыдущих расчетов
            ways[j] += ways[j-coin]
        }
    }
}
fmt.Println(ways[N]) // вывод: 5
```

Далее

Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»