Задача B. Подарки Орла

Автор и разработчик: Василий Гришков

Так как Леонид хочет собрать максимальное количество подарков, то ему выгоднее брать подарки меньшего размера. Получили идею для жадного алгоритма — на каждом шаге брать минимальный по размеру подарок.

Реализуем эту идею. Отсортируем размеры подарков по возрастанию. Заведем переменную $prefSum = 0$ — сумма размеров рассмотренных элементов. Теперь пройдемся по отсортированным размерам. Если на i-м шаге $prefSum + a_i \leqslant n$, то мы добавляем подарок в наш мешок. Иначе i-й подарок не помещается в мешок, тогда мы должны вывести ответ — нам удалось собрать $i-1$ подарок. Необходимо обратить внимание на то, что в большинстве языков программирования принята 0-индексация.

Итоговая асимптотика: $O(n \log n)$.

Задача C. Установка ёлки

Автор и разработчик: Матвей Кориненко

Для решения этой задачи необходимо найти квадрат со стороной x в прямоугольной таблице со сторонами n и m и максимальной суммой элементов внутри него.

Для того, чтобы написать быстрый код, решающий эту задачу, необходимо построить двумерный массив префиксных сумм на основе данного нам массива. Этот предпосчет осуществляется за время $O(nm)$.

Пусть $pref[i][j]$ — сумма элементов в прямоугольном сегменте таблицы с левым верхним углом в клетке $(1, 1)$ и правым нижним в (i, j). Тогда, если существует квадрат со стороной x с правым нижним углом в клетке (i, j), то сумма элементов в нем равна $sum = pref[i][j] - pref[i-x][j] - pref[i][j-x] + pref[i-x][j-x]$. Среди всех таких квадратов необходимо выбрать один с максимальной суммой и вывести координаты его левой верхней вершины. Чтобы полностью их перебрать все возможные квадраты внутри таблицы, достаточно пробежаться по всем точкам двумерного массива.

Итоговая асимптотика: $O(nm)$.

Задача D. Новогодний салют

Автор и разработчик: Ольга Поспелова

Заметим, что для $n = 2$ ответа не существует (есть всего одно возможное разбиение $1 + 1$, которое противоречит условию о различном количестве залпов).

Существует множество подходов к решению этой задачи, рассмотрим некоторые из них.

Подход 1 Первое число зафиксируем за 1, второе за $n - 1$, следующие будем подбирать таким образом, чтобы сумма первых i чисел давала произведение $n \times ... \times (n - i + 2)$. Тогда сумма всех n чисел будет равна $n \times ... \times 2 = n!$. Поймем, что за число мы будем ставить на i-м шаге. Это будет $(n \times ... \times (n-i+3)) \times (n-i+1)$. Заметим, что все числа в разложении на множители i-го выписанного числа являются делителями числа $n!$. Почему все выписанные числа будут различными остается в качестве домашнего задания читателю.

Подход 2 Посчитаем число $n!$ и запишем его в переменную x. Теперь будем выписывать на доску числа по такому правилу: Если число x делится на два, то поделим его на два, запишем вместо x и выпишем на доску. Иначе поделим его на 3, запишем его вместо x, на доску выпишем число $2x$. Так мы выпишем ровно $n-2$ числа. Оставшиеся два числа выпишем как $\frac{2x}{3}$ и $\frac{x}{3}$. Остается понять, почему мы всегда можем делить на 2 или 3. Для этого посчитаем степени 2 и 3 в разложении числа $n!$ на простые множители. Заметим, что при всех n сумма степеней 2 и 3 больше либо равна $n - 1$. Это означает, что мы можем поделить на 2 или 3 это число $n - 1$ раз, что и требуется в нашем решении.

Итоговая асимптотика: $O(q \times n)$.

Задача E. Новогодний стол

Автор и разработчик: Иван Девятериков

Рассмотрим стол как n подряд идущих мест и решим задачу для такого случая. Для этого воспользуемся динамическим программированием. Будем подсчитывать два массива: $take$ и $skip$. В i-й позиции каждого из массивов мы будем хранить максимальную возможную оценку, которую мы можем собрать на префиксе длины i, в $take_i$ — если берем i-е блюдо за B, в $skip_i$ — если берем i-е блюдо за A. Получается: $take_0 = b_0$, $skip_0 = a_0$, $take_i = b_i + skip_{i-1}$, $skip_i = a_i + \max(skip_{i-1}, take_{i-1})$. Мы научились решать задачу для прямоугольного стола длины n.

Вернемся к исходной задаче и научимся сводить ее к рассмотренной. Зафиксируем стол за первую позицию и растянем его в прямоугольный. Решим задачу аналогично ранее описанной. Заметим, что мы могли получить неверный ответ, если мы выбрали блюдо B в первой и последней позиции. Посчитаем динамику для двух различных случаев в разных массивах — когда мы берем или не берем блюдо B на первой позиции. Реализовать это можно, создав массивы $take$ и $skip$ для каждого случая. Тогда интересующий нас ответ — максимум из трех случаев:

1. значение $skip$, если мы взяли блюдо B на первой позиции,
2. значение $skip$, если мы не взяли блюдо B на первой позиции,
3. значение $take$, если мы не взяли блюдо B на первой позиции.

Итоговая асимптотика: $O(n)$.