

Информатика. 11 класс
Вариант 1

Задание 1. Натуральные числа x , y , z и w удовлетворяют четырём следующим уравнениям

$$\begin{aligned}x \wedge y &= 1057 \\ x \text{ xor } y &= 2566 \\ (x + y) \text{ xor } z &= 716 \\ z \vee w &= 4239\end{aligned}$$

Здесь $\wedge$ поразрядная конъюнкция, xor поразрядное исключающее или, $\vee$ поразрядная дизъюнкция (например, для чисел $12 \wedge 5 = 1100_2 \wedge 0101_2 = 0100_2 = 4$, $12 \text{ xor } 5 = 1100_2 \text{ xor } 0101_2 = 1001_2 = 9$). Ответы на вопросы должны быть аналитически обоснованы, однако все вычисления и переводы в другие системы счисления можно выполнять программой, не упоминая это в решении. Сколько различных четвёрок чисел будут являться решениями этой системы уравнений?

Ответ:

В двоичном виде числа 1057 и 2566 имеют вид 010000100001 и

101000000110. Там, где в первом числе (то есть поразрядной конъюнкции) 1, и в том, и в другом числе будет стоять единица в этом разряде. Если во втором числе стоит ноль в каком-то разряде, то это возможно, только если две единицы или два нуля, по цифре в первом числе это можно определить однозначно. Во втором числе 4 единиц, на этих 4 местах и нужно определить, что за цифры (очевидно, что если мы определим цифру на определённом разряде в x , то мы сможем однозначно определить цифру на этом же разряде в y). Так как на каждом из 4 мест есть два варианта для цифры, это даёт 16 вариантов для чисел x и y . Далее заметим, что в исключающем или у нас собраны все степени двойки, которые входят ровно в одно число, а в конъюнкции те, которые входят в оба числа. Значит, если мы сложим исключающее или с удвоенной конъюнкцией, то получим сумму x и y , это будет 4680. Подставим в третье уравнение. Из $4680 \text{ xor } z = 716$ можно по свойству исключающего или получить, что $z = 4680 \text{ xor } 716 = 1000010000100$. То есть z определяется однозначно. Наконец, по последнему уравнению имеем $1000010000100 \vee w = 1000010001111$. Там, где и в первом, и во втором числе в уравнении есть 1 (а она есть и там и там в 3 разрядах), в

числе w может быть любая цифра. Итого у w ещё есть 8 различных вариантов числа. Эти варианты перемножаются и получается 128 вариантов.

Критерии:

За полное без ошибок нахождение количества пар x и y 7 баллов, за нахождение z и w 10 баллов. За незначительные ошибки, не повлиявшие на ответ либо повлиявшие незначительно, такие как неучёт какой-либо из единиц, снимается два балла.

Задание 2. Для передачи сообщений по каналу связи используются последовательности из цифр от 0 до 9. Так как при передаче сообщений по этому каналу связи возможны помехи, было решено отправлять последовательности длиной, кратной 3, где цифры объединены в блоки по 3 цифры, где в каждом блоке сумма цифр чётная (тогда, если сумма цифр нечётная, значит, в блоке были помехи и его надо переслать). Пример такой последовательности 947 037 282. Какой длины должны быть последовательности, чтобы можно было передать как минимум 10^{15} различных последовательностей именно этой длины?

Пример: если передаются 0 и 1 блоками по две цифры, где каждый блок имеет сумму 1, то последовательностей длины 4 будет 4 (10 10, 10 01, 01 10, 01 01).

Ответ:

Чтобы сумма цифр была чётной, первые две цифры можно выбирать, как угодно, а последнюю мы выбираем чётную или нечётную в зависимости от того, была ли чётной сумма первых двух цифр. Если была, третью цифру выбираем чётную, и наоборот. Так как чётных и нечётных цифр поровну, то у нас получается $10 \cdot 10 \cdot 5$ комбинаций. Итого 500 вариантов. Так как все блоки не зависят друг от друга, то работает правило умножения и если блоков k , то последовательностей будет 500 в степени k , осталось лишь подобрать его так, чтобы это число было больше или равно 10^{15} . Легко понять, что $500^5 = 5^5 \cdot 10^{10} = 3125 \cdot 10^{10} < 10^{15} < 15625 \cdot 10^{12} = 500^6$. Значит, последовательности должны быть длины от 6 блоков и выше, то есть от 18 цифр.

Критерии:

За полностью правильный подсчёт количества различных блоков без правильного перемножения количества вариантов ставится максимум 10 баллов. От двух до пяти баллов снимается за различные ошибки с подсчётом вариантов.

Задание 3. Два игрока играют в следующую игру. На доске написано одно или несколько натуральных чисел. В свой ход игрок может стереть одно число на доске и написать вместо него три различных натуральных числа, которые в сумме дают стёртое. Например, можно стереть 8 и написать 4 3 1, или 5 2 1. Если число не представляется в виде суммы трёх различных натуральных чисел, то с ним никакой ход сделать нельзя. Проигрывает тот, кто не может сделать ход. Какой игрок, первый или второй, победит при правильной игре и какая у него должна быть стратегия в трёх следующих случаях: когда на доске изначально написано число 13, когда на доске написаны числа 30 и 39, когда на доске написано в самом начале два числа, 9 и 24.

Ответ:

1) Для начала выясним, что все числа до 5 включительно являются проигрышными. Действительно, с ними нельзя сделать никакого хода, так как они не раскладываются на сумму трёх различных натуральных чисел. Следовательно, все числа от 6 до 12 являются выигрышными, в них можно разложить число на три числа, меньшие 6. Число 13 так разложить не получится. При этом из 13 нельзя получить и два различных числа, большие или равные 6, чтобы оставить противнику два хода. Следовательно, это позиция выигрышна для второго игрока.

2) Выигрышным ходом здесь будет походить в 30 30 5 4. С 5 и 4 уже нельзя сделать никакого хода. В 30 же можно реализовать симметричную стратегию, повторяя ходы за соперником.

3) 24 надо разбить на 7 8 9. Дело в том, что все эти числа можно разбить только один раз, кроме 9. Но в 9 можно повторить ход за соперником, а 7 и 8 в сумме дают только два хода.

Критерии:

Каждый полностью правильно рассмотренный случай даёт 7 баллов. 2 балла снимается за правильно рассмотренный случай, в котором при этом не учтён какой-то ход противника, не совсем корректно описана стратегия победы. 4 балла за один случай ставится в случае, если приведена важная идея для стратегии, но не полностью описана сама стратегия.

Задание 4. В текстовом файле записана последовательность, состоящая из n натуральных чисел. Вася собирает самую большую подпоследовательность чисел из последовательности натуральных чисел. При этом ему нужно, чтобы каждое следующее число в этой подпоследовательности либо делило предыдущее нацело, либо делилось на

предыдущее нацело. Например, в последовательности 10 3 5 17 24 8 25 он может выбрать 10 5 25 или 3 24 8, обе подпоследовательности максимально возможной длины 3. Напишите программу, которая находит эту максимальную длину. На вход программе подаётся в первой строчке число n , затем во второй строке n чисел через пробел. В ответ выведите длину самой большой такой подпоследовательности.

Входные данные: подается число n от 1 до 20000, затем n чисел.

Пример:

Входные данные	Выходные данные
7 10 3 5 17 24 8 25	3
8 2 3 5 7 11 13 17 19	1

Ответ:

```
for nom_test in (1, 2, 3):
    with open(f'test11_1_4_{nom_test}.txt') as f:
        n = int(f.readline())
        arr = list(map(int, f.readline().split()))
        dp = [1] * n
        for i in range(n):
            mx = 1
            for j in range(i):
                if arr[i] % arr[j] == 0 or \
                    arr[j] % arr[i] == 0:
                    mx = max(mx, dp[j] + 1)
            dp[i] = mx
        print(max(dp))
```

Ответы на тесты.

Test11_1_4_1.txt: 16

Test11_1_4_2.txt: 174

Test11_1_4_3.txt: 2035

Критерии:

За правильный ответ на один тест 7 баллов, на два теста 15 баллов, на три теста 23 балла. 22 балла ставится, если на все тесты ответ отличается на 1 балл. 2 балла ставилось за частичное решение, когда находились последовательности немного меньше максимальной длины.

Задание 5. Бельчонок хочет взобраться на дерево, высотой от 2 до 5 метров включительно, на одной приглянувшейся ему аллее. Эта аллея имеет длину n , то есть состоит из n идущих на одинаковом друг от друга

расстоянии деревьев. При этом Бельчонок хочет, чтобы его дерево было строго выше левого и правого дерева, если они есть. К тому же, чтобы не загромождать ему вид, расстояние до ближайшего дерева с строго большей высотой должно быть максимально. В ответ надо записать номер места, на котором стоит дерево и расстояние до ближайшего большего дерева. Например, если высоты у деревьев будут 1 3 3 4 2 2 5 3 6 6, то самым подходящим будет четвёртое дерево, его расстояние до ближайшего дерева с высотой 5 будет 3. Если таких деревьев несколько, то надо выбирать дерево с наименьшей высотой, затем с наименьшим номером. Напишите программу, находящую этот ответ. Гарантируется, что в файле есть как минимум одно дерево ниже 5 и как минимум одно дерево выше 5. Учтите, что программа должна выдавать ответы на все тесты за время, не превышающее время олимпиады. Решения за $O(n^2)$ будут оцениваться неполным количеством баллов.

Входные данные: подается число n от 1 до 6000000, затем n чисел во второй строке, числа натуральные от 1 до 10.

Пример:

Входные данные	Выходные данные
10 1 3 3 4 2 2 5 3 6 6	4 3
10 1 5 1 1 2 10 1 10 2 1	2 4

```
for nom_test in (1, 2, 3):
    with open(f'test10_1_5_{nom_test}.txt') as f:
        n = int(f.readline())
        arr = list(map(int, f.readline().split()))
        ans = [0, 0]
        for floor in (2, 3, 4, 5):
            minl = [10**7] * n
            minr = [10**7] * n
            for i in range(1, n):
                if arr[i-1] > floor: minl[i] = 1
                else: minl[i] = minl[i - 1] + 1
            for i in range(n - 2, -1, -1):
                if arr[i+1] > floor: minr[i] = 1
                else: minr[i] = minr[i + 1] + 1
            max_dist = [0, 0]
            if max_dist[1] < min(minl[0], minr[0]) \
                and arr[0] == floor and arr[1]<arr[0]:
                max_dist = [1, min(minl[0], minr[0])]
            for i in range(1, n - 1):
                if max_dist[1] < min(minl[i], minr[i]) \
                    and arr[i] == floor and\
```

```
        arr[i]>arr[i+1] and arr[i]>arr[i-1]:
            max_dist = [i + 1, min(minl[i], minr[i])]
    if max_dist[1] < min(minl[-1], minr[-1]) and \
        arr[-1] == floor and arr[-2]<arr[-1]:
        max_dist = [n, min(minl[-1], minr[-1])]
    if max_dist[1] > ans[1]:
        ans = max_dist
    print(*ans)
```

Ответы на тесты.

Test11_1_5_1.txt: 856 6

Test11_1_5_2.txt: 39995 29997

Test11_1_5_3.txt: 417241 237216

Критерии:

За каждый правильно выполненный тест 8 баллов.

Информатика. 11 класс

Вариант 2

Задание 1. Натуральные числа x , y , z и w удовлетворяют четырём следующим уравнениям

$$x \wedge y = 1057$$

$$x \text{ xor } y = 2566$$

$$(x + y) \text{ xor } z = 2635$$

$$z \vee w = 6159$$

Здесь $\wedge$ поразрядная конъюнкция, xor поразрядное исключающее или, $\vee$ поразрядная дизъюнкция (например, для чисел $12 \wedge 5 = 1100_2 \wedge 0101_2 = 0100_2 = 4$, $12 \text{ xor } 5 = 1100_2 \text{ xor } 0101_2 = 1001_2 = 9$). Ответы на вопросы должны быть аналитически обоснованы, однако все вычисления и переводы в другие системы счисления можно выполнять программой, не упоминая это в решении. Сколько различных четвёрок чисел будут являться решениями этой системы уравнений?

Ответ:

В двоичном виде числа 1057 и 2566 имеют вид 010000100001 и 101000000110. Там, где в первом числе (то есть поразрядной конъюнкции) 1, и в том, и в другом числе будет стоять единица в этом разряде. Если во втором числе стоит ноль в каком-то разряде, то это возможно, только если две единицы или два нуля, по цифре в первом числе это можно определить однозначно. Во втором числе 4 единиц, на этих 4 местах и нужно определить, что за цифры (очевидно, что если мы определим цифру на определённом разряде в x , то мы сможем однозначно определить цифру на этом же разряде в y). Так как на каждом из 4 мест есть два варианта для цифры, это даёт 16 вариантов для чисел x и y . Далее заметим, что в исключающем или у нас собраны все степени двойки, которые входят ровно в одно число, а в конъюнкции те, которые входят в оба числа. Значит, если мы сложим исключающее или с удвоенной конъюнкцией, то получим сумму x и y , это будет 4680. Подставим в третье уравнение. Из $4680 \text{ xor } z = 2635$ можно по свойству исключающего или получить, что $z = 4680 \text{ xor } 716 = 1100000000011$. То есть z определяется однозначно. Наконец, по последнему уравнению имеем $1100000000011 \vee w = 1100000001111$. Там, где и в первом, и во втором числе в уравнении есть 1 (а она есть и там и там в 4 разрядах), в числе w может быть любая цифра. Итого у w ещё есть 16 различных вариантов числа. Эти варианты перемножаются и получается 256 вариантов.

Критерии:

За полное без ошибок нахождение количества пар x и y 7 баллов, за нахождение z и w 10 баллов. За незначительные ошибки, не повлиявшие на

ответ либо повлиявшие незначительно, такие как неучёт какой-либо из единиц, снимается два балла.

Задание 2. Для передачи сообщений по каналу связи используются последовательности из цифр от 0 до 9. Так как при передаче сообщений по этому каналу связи возможны помехи, было решено отправлять последовательности длиной, кратной 4, где цифры объединены в блоки по 4 цифры, где в каждом блоке сумма цифр чётная (тогда, если сумма цифр нечётная, значит, в блоке были помехи и его надо переслать). Пример такой последовательности 9472 1371 1281. Какой длины должны быть последовательности, чтобы можно было передать как минимум 10^{20} различных последовательностей?

Пример: если передаются 0 и 1 блоками по две цифры, где каждый блок имеет сумму 1, то последовательностей длины 4 будет 4 (10 10, 10 01, 01 10, 01 01).

Ответ:

Чтобы сумма цифр была чётной, первые три цифры можно выбирать, как угодно, а последнюю мы выбираем чётную или нечётную в зависимости от того, была ли чётной сумма первых трёх цифр. Если была, четвертую цифру выбираем чётную, и наоборот. Так как чётных и нечётных цифр поровну, то у нас получается $10 \cdot 10 \cdot 10 \cdot 5$ комбинаций. Итого 5000 вариантов. Так как все блоки не зависят друг от друга, то работает правило умножения и если блоков k , то последовательностей будет 5000 в степени k , осталось лишь подобрать его так, чтобы это число было больше или равно 10^{20} . Легко понять, что $5000^5 = 5^5 \cdot 10^{15} = 3125 \cdot 10^{15} < 10^{20} < 15625 \cdot 10^{18} = 500^6$. Значит, последовательности должны быть длины от 6 блоков и выше, то есть от 24 цифр.

Критерии:

За полностью правильный подсчёт количества различных блоков без правильного перемножения количества вариантов ставится максимум 10 баллов. От двух до пяти баллов снимается за различные ошибки с подсчётом вариантов.

Задание 3. Два игрока играют в следующую игру. На доске написано одно или несколько натуральных чисел. В свой ход игрок может стереть одно число на доске и написать вместо него три различных натуральных числа, которые в сумме дают стёртое. Например, можно стереть 8 и написать 4 3 1,

или 5 2 1. Если число не представляется в виде суммы трёх различных натуральных чисел, то с ним никакой ход сделать нельзя. Проигрывает тот, кто не может сделать ход. Какой игрок, первый или второй, победит при правильной игре и какая у него должна быть стратегия в трёх следующих случаях: когда на доске изначально написано число 13, когда на доске написаны числа 40 и 49, когда на доске написано в самом начале два числа, 10 и 23.

Ответ:

1) Для начала выясним, что все числа до 5 включительно являются проигрышными. Действительно, с ними нельзя сделать никакого хода, так как они не раскладываются на сумму трёх различных натуральных чисел. Следовательно, все числа от 6 до 12 являются выигрышными, в них можно разложить число на три числа, меньшие 6. Число 13 так разложить не получится. При этом из 13 нельзя получить и два различных числа, большие или равные 6, чтобы оставить противнику два хода. Следовательно, это позиция выигрышна для второго игрока.

2) Выигрышным ходом здесь будет походить в 40 40 5 4. С 5 и 4 уже нельзя сделать никакого хода. В 40 же можно реализовать симметричную стратегию, повторяя ходы за соперником.

3) 23 надо разбить на 6 7 10. Дело в том, что все эти числа можно разбить только один раз, кроме 10. Но в 10 можно повторить ход за соперником, а 7 и 6 в сумме дают только два хода.

Критерии:

Каждый полностью правильно рассмотренный случай даёт 7 баллов. 2 балла снимается за правильно рассмотренный случай, в котором при этом не учтён какой-то ход противника, не совсем корректно описана стратегия победы. 4 балла за один случай ставится в случае, если приведена важная идея для стратегии, но не полностью описана сама стратегия.

Задание 4. В текстовом файле записана последовательность, состоящая из n натуральных чисел. Вася собирает самую большую подпоследовательность чисел из последовательности натуральных чисел. При этом ему нужно, чтобы каждое следующее число в этой подпоследовательности либо делило предыдущее, либо делилось на предыдущее. Например, в последовательности 10 3 5 17 24 8 25 он может выбрать 10 5 25 или 3 24 8, обе подпоследовательности максимально возможной длины 3. Напишите программу, которая находит эту максимальную длину. На вход программе подаётся в первой строке число n ,

затем во второй строке n чисел через пробел. В ответ выведите длину самой большой такой подпоследовательности.

Входные данные: подается число n от 1 до 20000, затем n чисел.

Пример:

Входные данные	Выходные данные
7 10 3 5 17 24 8 25	3
8 2 3 5 7 11 13 17 19	1

Ответ:

```
for nom_test in (1, 2, 3):
    with open(f'test11_1_4_{nom_test}.txt') as f:
        n = int(f.readline())
        arr = list(map(int, f.readline().split()))
        dp = [1] * n
        for i in range(n):
            mx = 1
            for j in range(i):
                if arr[i] % arr[j] == 0 or \
                    arr[j] % arr[i] == 0:
                    mx = max(mx, dp[j] + 1)
            dp[i] = mx
        print(max(dp))
```

Ответы на тесты.

Test11_2_4_1.txt: 16

Test11_2_4_2.txt: 174

Test11_2_4_3.txt: 2035

Критерии:

За правильный ответ на один тест 7 баллов, на два теста 15 баллов, на три теста 23 балла. 22 балла ставится, если на все тесты ответ отличается на 1 балл. 2 балла ставилось за частичное решение, когда находились последовательности немного меньше максимальной длины.

Задание 5. Бельчонок хочет взобраться на дерево, высотой от 2 до 5 метров включительно, на одной приглянувшейся ему аллее. Эта аллея имеет длину n , то есть состоит из n идущих на одинаковом друг от друга расстоянии деревьев. При этом Бельчонок хочет, чтобы его дерево было строго выше левого и правого дерева, если они есть. К тому же, чтобы не

загораживать ему вид, расстояние до ближайшего дерева с строго большей высотой должно быть максимально. В ответ надо записать номер места, на котором стоит дерево и расстояние до ближайшего большего дерева. Например, если высоты у деревьев будут 1 3 3 4 2 2 5 3 6 6, то самым подходящим будет четвёртое дерево, его расстояние до ближайшего дерева с высотой 5 будет 3. Если таких деревьев несколько, то надо выбирать дерево с наименьшей высотой, затем с наименьшим номером. Напишите программу, находящую этот ответ. Гарантируется, что в файле есть как минимум одно дерево ниже 5 и как минимум одно дерево выше 5. Учтите, что программа должна выдавать ответы на все тесты за время, не превышающее время олимпиады. Решения за $O(n^2)$ будут оцениваться неполным количеством баллов.

Входные данные: подается число n от 1 до 6000000, затем n чисел во второй строке, числа могут быть от 1 до 10.

Пример:

Входные данные	Выходные данные
10 1 3 3 4 2 2 5 3 6 6	4 3
10 1 5 1 1 2 10 1 10 2 1	2 4

```
for nom_test in (1, 2, 3):
    with open(f'test10_1_5_{nom_test}.txt') as f:
        n = int(f.readline())
        arr = list(map(int, f.readline().split()))
        ans = [0, 0]
        for floor in (2, 3, 4, 5):
            minl = [10**7] * n
            minr = [10**7] * n
            for i in range(1, n):
                if arr[i-1] > floor: minl[i] = 1
                else: minl[i] = minl[i - 1] + 1
            for i in range(n - 2, -1, -1):
                if arr[i+1] > floor: minr[i] = 1
                else: minr[i] = minr[i + 1] + 1
            max_dist = [0, 0]
            if max_dist[1] < min(minl[0], minr[0]) \
                and arr[0] == floor and arr[1]<arr[0]:
                max_dist = [1, min(minl[0], minr[0])]
            for i in range(1, n - 1):
                if max_dist[1] < min(minl[i], minr[i]) \
                    and arr[i] == floor and\
```

```
        arr[i]>arr[i+1] and arr[i]>arr[i-1]:
            max_dist = [i + 1, min(minl[i], minr[i])]
    if max_dist[1] < min(minl[-1], minr[-1]) \
        and arr[-1] == floor and arr[-2]<arr[-1]:
        max_dist = [n, min(minl[-1], minr[-1])]
    if max_dist[1] > ans[1]:
        ans = max_dist
    print(*ans)
```

Ответы на тесты.

Test11_2_5_1.txt: 856 6

Test11_2_5_2.txt: 39995 29997

Test11_2_5_3.txt: 417241 237216

Критерии:

За каждый правильно выполненный тест 8 баллов.

Информатика. 11 класс

Вариант 3

Задание 1. Числа x , y , z и w удовлетворяют четырем следующим уравнениям:

$$x \wedge y = 1057$$

$$x \text{ xor } y = 2566$$

$$(x + y) \text{ xor } z = 3659$$

$$z \vee w = 7183$$

Здесь $\wedge$ поразрядная конъюнкция, xor поразрядное исключающее или, $\vee$ поразрядная дизъюнкция (например, для чисел $12 \wedge 5 = 1100_2 \wedge 0101_2 = 0100_2 = 4$, $12 \text{ xor } 5 = 1100_2 \text{ xor } 0101_2 = 1001_2 = 9$). Ответы на вопросы должны быть аналитически обоснованы, однако все вычисления и переводы в другие системы счисления можно выполнять программой, не упоминая это в решении. Сколько различных четвёрок чисел будут являться решениями этой системы уравнений?

Ответ:

В двоичном виде числа 1057 и 2566 имеют вид 010000100001 и 101000000110. Там, где в первом числе (то есть поразрядной конъюнкции) 1, и в том, и в другом числе будет стоять единица в этом разряде. Если во втором числе стоит ноль в каком-то разряде, то это возможно, только если две единицы или два нуля, по цифре в первом числе это можно определить однозначно. Во втором числе 4 единиц, на этих 4 местах и нужно определить, что за цифры (очевидно, что если мы определим цифру на определённом разряде в x , то мы сможем однозначно определить цифру на этом же разряде в y). Так как на каждом из 4 мест есть два варианта для цифры, это даёт 16 вариантов для чисел x и y . Далее заметим, что в исключающем или у нас собраны все степени двойки, которые входят ровно в одно число, а в конъюнкции те, которые входят в оба числа. Значит, если мы сложим исключающее или с удвоенной конъюнкцией, то получим сумму x и y , это будет 4680. Подставим в третье уравнение. Из $4680 \text{ xor } z = 3659$ можно по свойству исключающего или получить, что $z = 4680 \text{ xor } 3659 = 1100000000011$. То есть z определяется однозначно. Наконец, по последнему уравнению имеем $1100000000011 \vee w = 1100000001111$. Там, где и в первом, и во втором числе в уравнении есть 1 (а она есть и там и там в 5 разрядах), в числе w может быть любая цифра. Итого у w ещё есть 32 различных вариантов числа. Эти варианты перемножаются и получается 512 вариантов.

Критерии:

За полное без ошибок нахождение количества пар x и y 7 баллов, за нахождение z и w 10 баллов. За незначительные ошибки, не повлиявшие на

ответ либо повлиявшие незначительно, такие как неучёт какой-либо из единиц, снимается два балла.

Задание 2. Для передачи сообщений по каналу связи используются последовательности из цифр от 0 до 9. Так как при передаче сообщений по этому каналу связи возможны помехи, было решено отправлять последовательности длиной, кратной 3, где цифры объединены в блоки по 3 цифры, где в каждом блоке сумма цифр делится на 5 (тогда, если сумма цифр не делится на 5, значит, в блоке были помехи и его надо переслать). Пример такой последовательности 947 037 285. Какой длины должны быть последовательности, чтобы можно было передать как минимум 10^{15} различных последовательностей?

Пример: если передаются 0 и 1 блоками по две цифры, где каждый блок имеет сумму 1, то последовательностей длины 4 будет 4 (10 10, 10 01, 01 10, 01 01).

Ответ:

Чтобы сумма цифр делилась на 5, первые две цифры можно выбирать, как угодно, а последнюю мы можем выбрать из двух вариантов так, чтобы сумма трёх цифр делилась на 5 (если остаток у первых двух при делении на 5 равен 0, выбираем 0 или 5, если 1, то 4 или 9 и так далее). Итого, у нас получается $10 \cdot 10 \cdot 2$ комбинаций. Итого 200 вариантов. Так как все блоки не зависят друг от друга, то работает правило умножения и если блоков k , то последовательностей будет 200^k в степени k , осталось лишь подобрать его так, чтобы это число было больше или равно 10^{15} . Легко понять, что $200^6 = 2^6 \cdot 10^{12} = 64 \cdot 10^{12} < 10^{15} < 128 \cdot 10^{14} = 200^7$. Значит, последовательности должны быть длины от 7 блоков и выше, то есть от 21 цифры.

Критерии:

За полностью правильный подсчёт количества различных блоков без правильного перемножения количества вариантов ставится максимум 10 баллов. От двух до пяти баллов снимается за различные ошибки с подсчётом вариантов.

Задание 3. Два игрока играют в следующую игру. На доске написано одно или несколько натуральных чисел. В свой ход игрок может стереть одно число на доске и написать вместо него три различных натуральных числа, которые в сумме дают стёртое. Например, можно стереть 8 и написать 4 3 1,

или 5 2 1. Если число не представляется в виде суммы трёх различных натуральных чисел, то с ним никакой ход сделать нельзя. Проигрывает тот, кто не может сделать ход. Какой игрок, первый или второй, победит при правильной игре и какая у него должна быть стратегия в трёх следующих случаях: когда на доске изначально написано число 13, когда на доске написаны числа 38 и 47, когда на доске написано в самом начале два числа, 10 и 25.

Ответ:

1) Для начала выясним, что все числа до 5 включительно являются проигрышными. Действительно, с ними нельзя сделать никакого хода, так как они не раскладываются на сумму трёх различных натуральных чисел. Следовательно, все числа от 6 до 12 являются выигрышными, в них можно разложить число на три числа, меньшие 6. Число 13 так разложить не получится. При этом из 13 нельзя получить и два различных числа, большие или равные 6, чтобы оставить противнику два хода. Следовательно, это позиция выигрышна для второго игрока.

2) Выигрышным ходом здесь будет походить в 38 38 5 4. С 5 и 4 уже нельзя сделать никакого хода. В 38 же можно реализовать симметричную стратегию, повторяя ходы за соперником.

3) 25 надо разбить на 10 2 13. В двух числах 10 тогда можно реализовывать симметричную стратегию, повторяя за противником. 13, как мы выяснили, сама по себе проигрышная позиция, потому что если второй игрок сделает там какой-то ход, первый сразу же может его там победить. Также можно разбить на 7 8 10.

Критерии:

Каждый полностью правильно рассмотренный случай даёт 7 баллов. 2 балла снимается за правильно рассмотренный случай, в котором при этом не учтён какой-то ход противника, не совсем корректно описана стратегия победы. 4 балла за один случай ставится в случае, если приведена важная идея для стратегии, но не полностью описана сама стратегия.

Задание 4. В текстовом файле записана последовательность, состоящая из n натуральных чисел. Вася собирает самую большую подпоследовательность чисел из последовательности натуральных чисел. При этом ему нужно, чтобы каждое следующее число в этой подпоследовательности либо делило предыдущее, либо делилось на предыдущее. Например, в последовательности 10 3 5 17 24 8 25 он может выбрать 10 5 25 или 3 24 8, обе подпоследовательности максимально

возможной длины 3. Напишите программу, которая находит эту максимальную длину. На вход программе подаётся в первой строчке число n , затем во второй строке n чисел через пробел. В ответ выведите длину самой большой такой подпоследовательности.

Входные данные: подается число n от 1 до 20000, затем n чисел.

Пример:

Входные данные	Выходные данные
7 10 3 5 17 24 8 25	3
8 2 3 5 7 11 13 17 19	1

Ответ:

```
for nom_test in (1, 2, 3):
    with open(f'test11_1_4_{nom_test}.txt') as f:
        n = int(f.readline())
        arr = list(map(int, f.readline().split()))
        dp = [1] * n
        for i in range(n):
            mx = 1
            for j in range(i):
                if arr[i] % arr[j] == 0 or \
                    arr[j] % arr[i] == 0:
                    mx = max(mx, dp[j] + 1)
            dp[i] = mx
        print(max(dp))
```

Ответы на тесты.

Test11_3_4_1.txt: 16

Test11_3_4_2.txt: 174

Test11_3_4_3.txt: 2035

Критерии:

За правильный ответ на один тест 7 баллов, на два теста 15 баллов, на три теста 23 балла. 22 балла ставится, если на все тесты ответ отличается на 1 балл. 2 балла ставилось за частичное решение, когда находились последовательности немного меньше максимальной длины.

Задание 5. Бельчонок хочет взобраться на дерево, высотой от 2 до 5 метров включительно, на одной приглянувшейся ему аллее. Эта аллея имеет

длину n , то есть состоит из n идущих на одинаковом друг от друга расстоянии деревьев. При этом Бельчонок хочет, чтобы его дерево было строго выше левого и правого дерева, если они есть. К тому же, чтобы не загромождать ему вид, расстояние до ближайшего дерева с строго большей высотой должно быть максимально. В ответ надо записать номер места, на котором стоит дерево и расстояние до ближайшего большего дерева. Например, если высоты у деревьев будут 1 3 3 4 2 2 5 3 6 6, то самым подходящим будет четвёртое дерево, его расстояние до ближайшего дерева с высотой 5 будет 3. Если таких деревьев несколько, то надо выбирать дерево с наименьшей высотой, затем с наименьшим номером. Напишите программу, находящую этот ответ. Гарантируется, что в файле есть как минимум одно дерево ниже 5 и как минимум одно дерево выше 5. Учтите, что программа должна выдавать ответы на все тесты за время, не превышающее время олимпиады. Решения за $O(n^2)$ будут оцениваться неполным количеством баллов.

Входные данные: подается число n от 1 до 6000000, затем n чисел во второй строке, числа могут быть от 1 до 10.

Пример:

Входные данные	Выходные данные
10 1 3 3 4 2 2 5 3 6 6	4 3
10 1 5 1 1 2 10 1 10 2 1	2 4

Ответ:

```
for nom_test in (1, 2, 3):
    with open(f'test10_1_5_{nom_test}.txt') as f:
        n = int(f.readline())
        arr = list(map(int, f.readline().split()))
        ans = [0, 0]
        for floor in (2, 3, 4, 5):
            minl = [10**7] * n
            minr = [10**7] * n
            for i in range(1, n):
                if arr[i-1] > floor: minl[i] = 1
                else: minl[i] = minl[i - 1] + 1
            for i in range(n - 2, -1, -1):
                if arr[i+1] > floor: minr[i] = 1
                else: minr[i] = minr[i + 1] + 1
            max_dist = [0, 0]
            if max_dist[1] < min(minl[0], minr[0]) and \
                arr[0] == floor and arr[1] < arr[0]:
                max_dist = [1, min(minl[0], minr[0])]
```

```

for i in range(1, n - 1):
    if max_dist[1] < min(minl[i], minr[i]) and \
        arr[i] == floor and \
        arr[i] > arr[i+1] and arr[i] > arr[i-1]:
        max_dist = [i + 1, min(minl[i], minr[i])]
if max_dist[1] < min(minl[-1], minr[-1]) and \
    arr[-1] == floor and arr[-2] < arr[-1]:
    max_dist = [n, min(minl[-1], minr[-1])]
if max_dist[1] > ans[1]:
    ans = max_dist
print(*ans)

```

Ответы на тесты.

Test11_3_5_1.txt: 856 6

Test11_3_5_2.txt: 39995 29997

Test11_3_5_3.txt: 417241 237216

Критерии:

За каждый правильно выполненный тест 8 баллов.

Информатика. 11 класс

Вариант 4

Задание 1. Числа x , y , z и w удовлетворяют четырем следующим уравнениям:

$$x \vee y = 3623$$

$$x \text{ xor } y = 2566$$

$$(x + y) \text{ xor } z = 716$$

$$z \vee w = 4239$$

Здесь xor поразрядное исключающее или, $\vee$ поразрядная дизъюнкция (например, для чисел $12 \vee 5 = 1100_2 \vee 0101_2 = 1101_2 = 13$, $12 \text{ xor } 5 = 1100_2 \text{ xor } 0101_2 = 1001_2 = 9$). Ответы на вопросы должны быть аналитически обоснованы, однако все вычисления и переводы в другие системы счисления можно выполнять программой, не упоминая это в решении. Сколько различных четвёрок чисел будут являться решениями этой системы уравнений?

Ответ:

В двоичном виде числа 3623 и 2566 имеют вид 111000100111 и 101000000110. Там, где в первом числе (то есть поразрядной дизъюнкции) 0, и в том, и в другом числе будет стоять ноль в этом разряде. Если во втором

числе стоит ноль в каком-то разряде, то это возможно, только если две единицы или два нуля, по цифре в первом числе это можно определить однозначно. Во втором числе 4 единиц, на этих 4 местах и нужно определить, что за цифры (очевидно, что если мы определим цифру на определённом разряде в x , то мы сможем однозначно определить цифру на этом же разряде в y). Так как на каждом из 4 мест есть два варианта для цифры, это даёт 16 вариантов для чисел x и y . Далее заметим, что в исключающем или y нас собраны все степени двойки, которые входят ровно в одно число, а в дизъюнкции те, которые входят в хотя бы одно число. Значит, если мы удвоим дизъюнкцию и отнимем от неё исключающее или, то получим сумму x и y , это будет 4680. Подставим в третье уравнение. Из $4680 \text{ xor } z = 716$ можно по свойству исключающего или получить, что $z = 4680 \text{ xor } 716 = 1000010000100$. То есть z определяется однозначно. Наконец, по последнему уравнению имеем $1000010000100 \vee w = 1000010001111$. Там, где и в первом, и во втором числе в уравнении есть 1 (а она есть и там и там в 3 разрядах), в числе w может быть любая цифра. Итого у w ещё есть 8 различных вариантов числа. Эти варианты перемножаются и получается 128 вариантов.

Критерии:

За полное без ошибок нахождение количества пар x и y 7 баллов, за нахождение z и w 10 баллов. За незначительные ошибки, не повлиявшие на ответ либо повлиявшие незначительно, такие как неучёт какой-либо из единиц, снимается два балла.

Задание 2. Для передачи сообщений по каналу связи используются последовательности из цифр от 0 до 5. Так как при передаче сообщений по этому каналу связи возможны помехи, было решено отправлять последовательности длиной, кратной 3, где цифры объединены в блоки по 3 цифры, где в каждом блоке сумма цифр чётная и строго больше 2 (тогда, если сумма цифр нечётная, значит, в блоке были помехи и его надо переслать). Пример такой последовательности 213 040 523. Какой длины должны быть последовательности, чтобы можно было передать как минимум 10^{12} различных последовательностей?

Пример: если передаются 0 и 1 блоками по две цифры, где каждый блок имеет сумму 1, то последовательностей длины 4 будет 4 (10 10, 10 01, 01 10, 01 01).

Ответ:

Чтобы сумма цифр делилась на 2, первые две цифры можно выбирать, как угодно, а последнюю мы выберем из трёх вариантов (если сумма у первых двух чётная, выбираем любую чётную, если нечётная, то нечётную). Итого, у нас получается $6 \cdot 6 \cdot 3$ комбинаций. Это 108 вариантов. Вычтем из

них те, где сумма цифр равна 0 или 2. Их всего 7, и у нас остаётся 101 вариант. Так как все блоки не зависят друг от друга, то работает правило умножения и если блоков k , то последовательностей будет 101 в степени k , осталось лишь подобрать его так, чтобы это число было больше или равно 10^{12} . Легко понять, что $101^5 = 1051010050 < 10^{12} = 100^6 < 101^6$. Значит, последовательности должны быть длины от 6 блоков и выше, то есть от 18 цифр.

Критерии:

За полностью правильный подсчёт количества различных блоков без правильного перемножения количества вариантов ставится максимум 10 баллов. От двух до пяти баллов снимается за различные ошибки с подсчётом вариантов.

Задание 3. Два игрока играют в следующую игру. На доске написано одно или несколько натуральных чисел. В свой ход игрок может стереть одно число на доске и написать вместо него три различных натуральных числа, которые в сумме дают стёртое. Например, можно стереть 8 и написать 4 3 1, или 5 2 1. Если число не представляется в виде суммы трёх различных натуральных чисел, то с ним никакой ход сделать нельзя. Проигрывает тот, кто не может сделать ход. Какой игрок, первый или второй, победит при правильной игре и какая у него должна быть стратегия в трёх следующих случаях: когда на доске изначально написано число 13, когда на доске написаны числа 50 и 63, когда на доске написано в самом начале два числа, 10 и 22.

Ответ:

1) Для начала выясним, что все числа до 5 включительно являются проигрышными. Действительно, с ними нельзя сделать никакого хода, так как они не раскладываются на сумму трёх различных натуральных чисел. Следовательно, все числа от 6 до 12 являются выигрышными, в них можно разложить число на три числа, меньшие 6. Число 13 так разложить не получится. При этом из 13 нельзя получить и два различных числа, большие или равные 6, чтобы оставить противнику два хода. Следовательно, это позиция выигрышна для второго игрока.

2) Выигрышным ходом здесь будет походить в 50 50 6 7. С 6 и 7 можно сделать по одному ходу, то есть это чётное число ходов. В 50 же можно реализовать симметричную стратегию, повторяя ходы за соперником.

3) 22 надо разбить на 9 7 6. Числа 7 и 6 дают ровно по одному ходу, потому что это чётное число ходов, не влияющее на то, кто победит. 9 и 10 же

можно разбить как за один ход, так и за два. Если второй игрок разбивает одно из этих чисел полностью на неразбиваемые, первый делает то же самое, но с другим числом. Если же второй игрок разбивает их так, что остаётся ещё один ход (например, 6 2 1 или 7 2 1), то мы симметрично делаем также со вторым.

Критерии:

Каждый полностью правильно рассмотренный случай даёт 7 баллов. 2 балла снимается за правильно рассмотренный случай, в котором при этом не учтён какой-то ход противника, не совсем корректно описана стратегия победы. 4 балла за один случай ставится в случае, если приведена важная идея для стратегии, но не полностью описана сама стратегия.

Задание 4. В текстовом файле записана последовательность, состоящая из n натуральных чисел. Вася собирает самую большую подпоследовательность чисел из последовательности натуральных чисел. При этом ему нужно, чтобы каждые два соседних числа в этой подпоследовательности давали в сумме квадрат или куб натурального числа. Например, в последовательности 10 3 15 17 6 10 12 он может выбрать 10 15 12 или 3 6 10, обе подпоследовательности максимально возможной длины 3. Напишите программу, которая находит эту максимальную длину. На вход программе подаётся в первой строчке число n , затем во второй строке n чисел через пробел. В ответ выведите длину самой большой такой подпоследовательности.

Входные данные: подается число n от 1 до 20000, затем n чисел.

Пример:

Входные данные	Выходные данные
7 10 3 15 17 6 10 12	3
8 100 1 2 4 9 1 1 1	1

Ответ:

```
cubes = set([i**3 for i in range(2, 10**3+1)] + \
            [i**2 for i in range(1, 4*10**4)])
for nom_test in (1, 2, 3):
    with open(f'test11_4_4_{nom_test}.txt') as f:
        n = int(f.readline())
        arr = list(map(int, f.readline().split()))
```

```

dp = [1] * n
for i in range(n):
    mx = 1
    for j in range(i):
        x, y = arr[i], arr[j]
        if (x+y) in cubes:
            mx = max(mx, dp[j] + 1)
    dp[i] = mx
print(max(dp))

```

Ответы на тесты: 3 19 151

Test11_4_4_1.txt: 3

Test11_4_4_2.txt: 19

Test11_244_3.txt: 151

Критерии:

За правильный ответ на один тест 7 баллов, на два теста 15 баллов, на три теста 23 балла. 22 балла ставится, если на все тесты ответ отличается на 1 балл. 2 балла ставилось за частичное решение, когда находились последовательности немного меньше максимальной длины.

Задание 5. Бельчонок хочет взобраться на дерево на одной приглянувшейся ему аллее. Эта аллея имеет длину n , то есть состоит из n идущих подряд на одинаковом расстоянии друг от друга деревьев. При этом Бельчонок не хочет, чтобы рядом с его деревом были деревья такие же по высоте или выше, чтобы они не загораживали ему вид. Поэтому он хочет, чтобы расстояние до ближайшего дерева с такой же или большей высотой было максимально. В ответ надо записать номер места, на котором стоит дерево и это расстояние. Например, если высоты у деревьев будут 1 3 3 4 2 2 5 3 6 6, то самым подходящим будет четвёртое дерево, его расстояние до ближайшего дерева с высотой 5 будет 3. Если таких деревьев несколько, то надо выбирать дерево с наименьшей высотой, затем с наименьшим номером. Гарантируется, что в файлах есть как минимум два дерева каждой высоты.

Входные данные: подается число n от 1 до 6000000, затем n чисел во второй строке, числа натуральные от 1 до 9.

Пример:

Входные данные	Выходные данные
10 1 3 3 4 2 2 5 3 6 6	4 3

10 1 7 1 1 2 9 1 9 2 1	2 4
---------------------------	-----

Ответ:

```

for nom_test in (1, 2, 3):
    with open(f'test11_1_5_{nom_test}.txt') as f:
        n = int(f.readline())
        arr = list(map(int, f.readline().split()))
        ans = [0, 0]
        for floor in range(1, 10):
            minl = [10**7] * n
            minr = [10**7] * n
            for i in range(1, n):
                if arr[i-1] >= floor: minl[i] = 1
                else: minl[i] = minl[i - 1] + 1
            for i in range(n - 2, -1, -1):
                if arr[i+1] >= floor: minr[i] = 1
                else: minr[i] = minr[i + 1] + 1
            max_dist = [0, 0]
            for i in range(1, n - 1):
                if max_dist[1] < min(minl[i], minr[i]) and\
                    arr[i] == floor:
                    max_dist = [i + 1, min(minl[i], minr[i])]
            if max_dist[1] > ans[1]:
                ans = max_dist
        print(*ans)

```

Ответы на тесты.

Test11_4_5_1.txt: 608 36

Test11_4_5_2.txt: 9977 71

Test11_4_5_3.txt: 41566 55

Критерии:

За каждый правильно выполненный тест 8 баллов.