

Всероссийская олимпиада школьников по информатике
Вологодская область, 2023-24 учебный год
II (муниципальный) этап
7 - 8 классы

Методические рекомендации по разбору задач

Задача 1. Ромбы

В первом вопросе ответ 45 несложно получить вручную, построив ещё один ромб и кусочек следующего.

Во втором вопросе можно заметить, что размер каждого следующего ромба на 4 больше предыдущего. Тогда суммарное количество чисел в центральной клетке и первых 9 ромбах равно $1 + 4 * (1 + 2 + \dots + 9) = 181$, и тогда следующее число 182 будет ответом.

В третьем вопросе сначала аналогичным образом найдём количество чисел в центральной клетке и первых 11 ромбах --- это будет 265. Теперь прибавим к нему длину стороны 12-го ромба, и ответом будет $265 + 13 = 278$.

В четвёртом вопросе найдём сначала номер ромба. Заметим, что если бы второе число равнялось 0, то клетка лежала бы на ромбе с номером 12345. Каждое увеличение y на 1 увеличивает и номер ромба на 1, поэтому номер ромба равен $12345 + 54321 = 66666$. Количество чисел в центральной клетке и всех предыдущих ромбах равно $1 + 4 * (1 + 2 + 3 + \dots + 66665)$. Используя формулу суммы арифметической прогрессии, получаем $1 + 4 * (1 + 66665) / 2 * 66665 = 8888577781$. Осталось добавить смещение по стороне последнего ромба, оно равно 12346. Итого получается ответ 8888590127.

Заметим, что можно вывести общую формулу для получения ответа: для неотрицательных x и y ответ равен $2*(x+y)*(x+y-1)+x+2$.

Все ответы: 45 182 278 8888590127

Задача 2. Программа для робота

Методом пристального взгляда либо путём написания программы перебора вариантов можно найти, например, такое решение из 8 команд: D3 R3 L1 U7 R5 U3 R2 D7. Возможны и другие правильные варианты.

Решения меньше чем из 8 команд существовать не должно, это доказано с помощью программы, выполняющей полный перебор всех более коротких программ.

Заметим, что оптимальное решение придумать не очень просто. Однако, совсем несложно получить программу из 10 команд, набирающую 82 балла. Вначале разберёмся с клетками на границах поля: U7 R7 D7. Теперь посмотрим на третью строку снизу. Поскольку последней командой была D7, то для доведения до выхода робота из любой клетки в этой строке достаточно добавить команды:

R3 L1 U2 R1 U3 R3 D7. Несложно убедиться, что для всех остальных клеток поля робот при этом тоже дойдёт до выхода.

Задача 3. Количество слов

В первом вопросе можно просто перечислить все 6 слов: аос, асо, оас, оса, сао, соа.

Во втором вопросе заметим, что в каждую из трёх позиций можно поставить любую из трёх букв. Итого $3*3*3=27$ слов.

В третьем вопросе переберём, сколько раз будет встречаться в слове буква 'а'. Пусть p – количество букв 'а'. Расставить p букв в n позиций можно $C(n, p)$ способами, где $C(n, p)$ – количество сочетаний из n по p . Дальше для каждого p переберём количество букв 'о'. Пусть q – количество букв 'о'. Их нужно расставить в $(n - p)$ позиций – это $C(n-p, q)$ способов. И нужно ещё смотреть, чтобы количество оставшихся букв не превышало $K=3$. Получаем такую сумму:

$$\begin{aligned} &C(4, 0) * (C(4, 1) + C(4, 2) + C(4, 3)) + \\ &C(4, 1) * (C(3, 0) + C(3, 1) + C(3, 2) + C(3, 3)) + \\ &C(4, 2) * (C(2, 0) + C(2, 1) + C(2, 2)) + \\ &C(4, 3) * (C(1, 0) + C(1, 1)). \end{aligned}$$

Напомним, что число сочетаний можно сосчитать по формуле:

$$C(n, m) = n! / (m! * (n - m)!).$$

Вычисление по этой формуле даёт ответ 78.

Для ответа на 4-й вопрос можно написать программу, которая выполнит вычисления аналогично написанному выше ручному расчёту, либо воспользоваться методом динамического программирования. Подробности смотрите в разборе похожей задачи на программирование у 9-11 классов.

Все ответы: 6 27 78 3093092574.

Задача 4. Пирожные

Можно сразу считать, что все пирожные разрезаны, тогда у нас $2A$ половинок.

Для решения первой подзадачи можно в цикле добавлять единицу к A , пока число $2A$ не станет делиться на D .

Для полного решения заметим, что нам нужно найти такое минимальное X , что $2 * (A + X)$ делится на D . Если D – чётное, то сократим эту двойку, поделив D на 2, а если нечётное, то просто её отбросим. Мы пришли к задаче: найти минимальное X , такое что $A + X$ делится на D . Его можно найти как $(D - A \bmod D) \bmod D$. Тогда ответом будет $A + (D - A \bmod D) \bmod D$.

Пример решения на Python:

```
d = int(input())
a = int(input())
```

```

if d % 2 == 0:
    d //= 2
print(a + (d - a % d) % d)

```

Задача 5. Поездки на курсы

В первой подзадаче можно создать массив дней. Для каждой поездки проходим в цикле от d_1 до d_2 и заносим номер города в элемент массива дней с соответствующим индексом. Если же там уже занесён другой город, то отмечаем это в отдельном массиве дубликатов. Затем остаётся только выбрать самый первый дубликат. Пример такого решения на Python:

```

n = int(input())
days = [0] * 1001
dup = [False] * 1001
for _ in range(n):
    d1, d2, c = map(int, input().split())
    for i in range(d1, d2 + 1):
        if days[i] != c and days[i] != 0:
            dup[i] = True
        days[i] = c
for i in range(1, 1001):
    if dup[i]:
        print(i)
        break
else:
    print(0)

```

Полное решение выглядит так. Заметим, что поскольку интервалы отсортированы, массивы для решения не нужны. Поддерживаем текущий интервал $\langle \text{left}, \text{right}, \text{city} \rangle$. Читаем очередной интервал $\langle d_1, d_2, c \rangle$. Если $d_1 > \text{right}$, то просто заменяем текущий интервал на новый. Иначе сравниваем города. Если города одинаковы, то обновляем правую границу текущего интервала: $\text{right} = \max(\text{right}, d_2)$. Если же города разные, то ответом будет d_1 .

Пример решения на Python:

```

n = int(input())
left = right = city = ans = 0
for _ in range(n):
    d1, d2, c = map(int, input().split())
    if d1 > right:
        left = d1
        right = d2
        city = c
    elif c == city:
        right = max(right, d2)
    else:
        ans = d1
        break
print(ans)

```