

## Олимпиада для 6-8 классов 2024 год.

### Разбор заданий.

#### Задача 1. «Покраска»

Подгруппа 1.  $N = 1$ . В данной группе может быть только 1 тест: 1 1 1, который означает, что у нас полоска длины 1, и в эту клетку изначально были помещены обе капли. Поскольку вся полоска уже является окрашенной, то ответ на этот тест – 0.

Интересный факт. Среди всех участников олимпиады у двух человек по данной задаче 95 баллов из-за того, что их решение выдаёт верный результат на всех тестах, кроме этого.

Подгруппа 2.  $N = 2$ . В данной группе ответом может быть либо 0, если ребята изначально окрасили различные клетки, либо 1, если они изначально окрасили одну и ту же клетку.

Подгруппа 3.  $N = 3$ . Данная группа была сделана для подсказки того, что проверить достижения двух концов было недостаточно. Если решение не проходило третью подгруппу, то участнику следовало найти тест 3 1 3, в котором на полоске длины 3 были окрашены первая и последняя клетки, но ответ при этом не 0.

Подгруппа 4.  $A = B$ . В данной группе необходимо было проверить, до какого края заполнение произойдёт медленнее – до начала или до конца. Из двух чисел следовало выбрать наибольшее.

Подгруппа 5.  $N \leq 10^5$ . Данная группа заходила, если участник хотел завести массив, в котором происходит моделирование процесса заполнения ячеек.

**Полное решение.** Можно заметить, что всего существует 3 промежутка, которые необходимо закрасить: от 1 до первой закрашенной клетки, между двумя клетками, а также от второй закрашенной клетки до конца полоски.

Для того, чтобы закрасить первый промежуток, требуется  $(a - 1)$  секунд, чтобы закрасить третий промежуток, требуется  $(n - b)$  секунд, а чтобы закрасить второй промежуток, необходимо  $(b - a)/2$  секунд, так как процесс закрашивания происходит с двух сторон. Затем из полученных трёх чисел необходимо вывести наибольшее.

## Задача 2. «Вечеринка»

Подгруппа 1.  $N = 0$ . В данной группе не было одиночных гостей. Необходимо было проверить, какое максимальное количество сока разного типа может выпить пара. Можно было заметить, что если количество сока какого-то вида нечётное, то последний стакан не будет выпит никогда.

Подгруппа 2.  $M = 0$ . В данной группе не было парных гостей, поэтому каждый сок мог бы быть теоретически выпит при наличии достаточного количества гостей, поэтому ответ – максимум из количества костей и суммарного количества оставшегося сока.

Подгруппа 3. Сок только одного вида. В данной группе необходимо было сразу проверить, какое максимальное количество выпьют пары, а затем одиночки. Небольшой «подсказкой» служило то, что оптимальный порядок проверки был указан в порядке следования подгрупп. Первая подгруппа указывала на проверку пар, а вторая – на проверку одиночных гостей.

Подгруппа 4. Все числа не превышают 1000. В данной подгруппе допускалось решение, когда сок отнимался в цикле `while` до тех пор, пока оставались гости или сок.

**Полное решение.** Выгоднее всего сразу раздать лимонад парам, а затем одиночкам. Поэтому сразу вычисляем, сколько пар выпьют сока одного типа, затем, сколько пар выпьют сока другого типа. Отнимем эти значения из количества сока. После этого к ответу прибавим минимальное из (количество одиночных гостей, суммарное количество оставшегося сока).

## Задача 3. «Лягушка»

Подгруппа 1.  $R \leq 10$ . В данной подгруппе будем считать, что величина  $x > 0$  (если  $x = 0$ , это решение других подгрупп).

Лягушка сделает не более 4 прыжков. Даже при самых минимальных начальных значениях  $k = 1$ ,  $x = 1$ , длины прыжков будут как минимум 1,2,3,4, что в сумме даёт 10. Можно было вручную просчитать 4 первых прыжка лягушки и для каждого из них проверить, попадает ли значение в промежуток ответа.

Подгруппа 2.  $L = R$ ,  $x = 0$ . В данной подгруппе длина прыжка не увеличивается и озеро состоит из одной точки. Ответ Yes будет в том случае, если  $L$  делится

нацело на  $k$ . Если же не делится, то необходимо было найти ближайшую точку, строго большую  $L$ , которая делится на  $k$ . Это можно сделать при помощи операций  $\text{div/mod}$ , например,  $L - (L \bmod k) + k$ .

Подгруппа 3.  $L = R$ . Данная подгруппа близка к полному решению, она требует применения цикла, но в данном цикле чуть проще, чем в полном решении, проверка того, находится ли лягушка в озере. В данной проверке не требовалось применять логическое условие `and`.

Подгруппа 4.  $x = 0$ . Данная подгруппа также близка к полному решению. В ней не было необходимости увеличивать длину прыжка внутри цикла

**Полное решение.** Заведём две переменные, одну, например,  $s$  – отвечающую за сумму всех элементов последовательности, а другую, например,  $\text{cur}$  – отвечающую за длину текущего прыжка.

Изначально  $s = 0$ ,  $\text{cur} = k$ .

Затем в цикле `for` (либо `while`, кому как удобнее) будем делать 3 следующие операции по очереди:

- 1) Прибавлять к сумме длину текущего прыжка
- 2) Выполнять проверку, не достигли ли мы точки прекращения действия цикла (ситуации, когда  $s$  находится либо между левой и правой точками, принадлежащим границам озера, либо превышает правую точку). Как только мы такой ситуации достигли – выводим YES либо NO в зависимости от того, где именно мы остановились, а на следующей строке – полученную сумму.
- 3) Увеличивать длину текущего прыжка на  $x$ .

#### Задача 4. «Поленья»

Подгруппа 1.  $N = 1$ . Нам дано одно полено длины  $x$ . Если оно нечётное – ответ 1  $x$ , иначе – рубим полено на две части, ответ 2  $x/2$ .

Подгруппа 2.  $N = 2$ . Самая интересная группа для участников, не знакомых с понятием массива. Следует рассмотреть ситуации, когда нам даны два полена длины  $x_1, x_2$ . Возможные варианты:

- 1) оба нечётные и различные – ответ 1  $\min(x_1, x_2)$
- 2) оба нечётные и совпадают – ответ 2  $x_1$
- 3) большее полено при разрубании становится равным меньшему – ответ 3  $\min(x_1, x_2)$

4) одно полено чётное, другое нечётное, но чётное при разрубании не становится равным другому – ответ  $2 \text{ чётное}/2$

5) оба полена чётные, разной длины, но большее при разрубании не становится равным меньшему – ответ  $2 \min(x1, x2)/2$

6) оба полена чётной длины и равны между собой – ответ  $4 \times 1$ .

Подгруппа 3. Длины поленьев либо 1, либо 2. Необходимо было подсчитать, сколько поленьев длины 2. Обозначим это число за cnt. Тогда ответом будет  $(n + cnt) / 1$ , так как каждое полено длины 2 будет расколото на поленья длины 1.

Подгруппа 4. Длины поленьев равны 1, 2 или 3. Также можно было подсчитать количество поленьев каждой длины. Обозначим их количества соответственно cnt1, cnt2, cnt3. Единичных поленьев можно получить  $cnt1 + 2 \times cnt2$ , длины 2 и 3 соответственно их изначальному количеству. Из всех вариантов следовало выбрать тот, который подходит под критерии условия задачи.

Подгруппа 5.  $N \leq 1000$ . В данной подгруппе можно было применить идеи квадратичной сортировки либо вложенных друг в друга циклов, проходя каждый раз по всему массиву от начала до конца для подсчёта различных кандидатов для ответа.

Полное решение. Применим идею, сходную с использованием сортировки подсчётом ([https://ru.wikipedia.org/wiki/Сортировка\\_подсчётом](https://ru.wikipedia.org/wiki/Сортировка_подсчётом)). Заведём массив **a**, в котором каждый элемент  $a[i]$  будет означать количество элементов  $i$  в исходном массиве. Затем пройдем последовательно по массиву **a** от 1 до миллиона. Затем каждый элемент  $i$  можно представить либо как маленькое полено, в этом случае формула пересчёта выглядит как  $a[i] + a[i \times 2] \times 2$ , либо как большое полено, в этом случае формула пересчёта выглядит как  $a[i] \times 2 + a[i \text{ div } 2]$ , если  $i$  – чётное. Из всех выражений необходимо взять наибольшее. Среди равных наибольших выбрать наименьшее  $i$ .

### Задача 5. «Выручка»

Подзадача 1.  $n = 1, m = 1$ . Требовалось определить, является ли единственное заданное число больше, меньше или равно 0.

Подзадача 2.  $m = 1, L = R$ . В данной подзадаче количество элементов могло быть большим, поэтому требовалось завести массив, с помощью которого можно определить знак у определённого элемента.

Подзадача 3.  $N = 2$ . Так как в данной подзадаче всего 2 элемента в массиве, то запросы могут быть одного из 3 видов: 1 1, 2 2, 1 2. Каждый из них можно было предподсчитать вручную.

Подзадача 4. Идентичная подзадаче 2, но запросов может быть много. Требуется цикл, чтобы отвечать на запросы.

Подзадача 5. В данной подзадаче массив длиной не более 10 и все элементы по модулю не превосходят 10. В данной подзадаче можно было «почестному» перемножать числа для каждого запроса во внутреннем цикле.

Подзадача 6.  $n, m \leq 1000$ . В данной подзадаче перемножать числа уже нельзя из-за переполнения. Однако чисто теоретически некоторые тесты могли пройти у тех участников, которые пишут на языке python из-за встроенной в данный язык длинной арифметики.

Подзадача 7.  $n \leq 1000, m \leq 10^5$  В данной подзадаче перемножение чисел «почестному» не зайдёт ни на python, ни на любом другом языке, даже если написать длинную арифметику вручную. Здесь необходимо было заметить, что результат не изменится, если каждое положительное число заменить на 1, каждое отрицательное – на -1, а нули оставить как есть. В таком случае можно аккуратно воспроизвести все запросы, перемножая данные числа.

Асимптотика  $O(n * m)$  в данной группе позволяет получить по ней баллы.

**Полное решение.** Создадим два массива префиксных сумм *pref\_zero* и *pref\_neg* – для нулевых и отрицательных элементов. *pref\_zero[i]* будет означать количество нулей на промежутке от 1 до *i*, *pref\_neg[i]* будет означать количество отрицательных элементов на промежутке от 1 до *i*.

Каждый запрос произведения, который задаётся числами *L*, *R* можно вычислять за  $O(1)$ . Для этого необходимо вначале узнать, есть ли на этом промежутке хотя бы один ноль. Для этого необходимо выполнить операцию *pref\_zero[R] – pref\_zero[L-1]*. Если есть – сразу выводим знак равенства и переходим к следующему запросу.

Если нулей нет, то необходимо вычислить количество отрицательных элементов, для этого необходимо выполнить операцию *pref\_neg[R] – pref\_neg[L – 1]*. Если количество отрицательных элементов чётное, то необходимо вывести знак «больше», иначе – знак «меньше».