Разбор задач

Задача 1. Делимость на 9

В этом задании было несколько вариантов, различающихся числами на данных наборах карточек. При этом принципы построения каждого набора карточек и способ нахождения ответа совпадали во всех вариантах.

Разберём один из вариантов. Во всех пунктах на карточках нет цифры 9, поэтому ответ нужно получить, выбрав минимальное число карточек, дающих в сумме 9. Затем записать цифры на этих карточках в порядке неубывания.

а) На карточках записаны числа 1, 3, 5, 6, 7.

Есть две карточки, дающие в сумме 9: $3 + 6 = 9$. Ответ: 36.

б) На карточках записаны числа 7, 3, 1, 2, 6, 5.

Из этих карточек число 9 можно составить двумя способами: $9 = 7+2 = 3+6$. Чтобы получить минимальное число, нужно выбрать на первое место минимальную цифру, в данном случае можно выбрать цифру 2. Ответ: 27.

в) На карточках записаны числа 2, 2, 4, 3, 3, 3, 1, 1, 1, 1.

Несложно видеть, что здесь нет двух цифр, дающих в сумме 9, поэтому будем пытаться составить число 9 из трёх цифр. Цифру 1 использовать не получится, т.к. нет двух цифр 4. А цифру 2 использовать можно, $9 = 2 + 3 + 4$. Ответ: 234.

г) На карточках записаны числа 6, 6, 8, 4, 4, 2, 2, 4, 6, 2.

Нельзя составить 9 из карточек, попробуем составить число 18. Это можно сделать используя три цифры. Ответ: 468.

д) На карточках записаны числа 4, 8, 4, 8, 4, 8, 4, 8, 4, 8, 4, 8.

Попробуем перебрать, например, количество цифр 4 в записи ответа. Если не использовать 4 совсем, а использовать только цифры 8, то чтобы составить кратное 9 понадобится девять цифр 8.

Если взять одну цифру 4, то $4 + 8 + 8 + 8 + 8 = 36$.

Если взять две цифры 4, то $4 + 4 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 = 72$.

Если взять три цифры 4, то $4 + 4 + 4 + 8 + 8 + 8 = 36$.

Перебирая дальше количество взятых цифр 4 увидим, что минимальный ответ будет 48888.

Задача 2. Муравей и коробка

У параллелепипеда 8 вершин, и в каждую вершину входит 3 ребра. Представим себе маршрут муравья. Кроме первой и последней вершины маршрута, в каждую вершину муравей должен несколько раз прийти и столько же раз уйти, значит, в каждой промежуточной вершине маршрута заканчивается чётное число рёбер маршрута муравья.

Поскольку в каждую вершину входит ровно 3 ребра параллелепипеда, и по всем этим рёбрам муравей должен пройти, по некоторым рёбрам муравей должен пройти дважды, чтобы во всех промежуточных вершинах заканчивалось чётное число рёбер пути. Минимальное число рёбер, которое нужно добавить, равно 3, потому что число промежуточных вершин равно 6.

Из трёх видов рёбер наименьшее — это ребро a, поэтому попробуем составить маршрут, в котором все рёбра коробки встречаются по одному разу, кроме трёх рёбер вида a, которые встречаются по два раза.

Такой путь можно построить. Сначала обойдём грань размером $b \times c$ по четырём рёбрам: $bcbc$. Затем по ребру a перейдём на другую грань $b \times c$. Также обойдём эту грань, при этом из каждой новой вершины, в которую пришёл муравей, будет идти ребро a к той грани, которую мы обошли вначале. По этому ребру пройдёмся два раза подряд: в одном и в другом направлении. Вторая часть маршрута будет иметь вид *baacaabaac*.

Ответ: *bcbcabaacaabaac*.

Задача 3. Морской бой

У этой задачи было несколько вариантов, отличающихся начальным расположением кораблей. Во всех вариантах лучшее решение содержит 6 выстрелов.

Мы можем сделать 10 выстрелов отдельно по каждом кораблю. Чтобы уменьшить количество выстрелов, нужно выбирать моменты, когда два двигающихся навстречу корабля оказываются в одной точке. Несложно заметить, что «пересечения» кораблей будут зачастую происходить одновременно, поэтому необходимо изучить, в какие моменты будут происходить пересечения кораблей, т.к. производить выстрелы нужно в разные моменты времени.

Будем обозначать корабли их начальной координатой, они различны у всех кораблей. Направо движутся корабли 10, 12, 20, 32, 42, налево движутся корабли 18, 26, 28, 30, 48.

Составим таблицу, в которой в строках будут корабли, двигающиеся направо, в столбцах — корабли, двигающиеся налево, а на пересечении строки и столбца — момент времени, когда эти корабли пересекутся.

Корабли	$\leftarrow 18$	$\leftarrow 26$	$\leftarrow 28$	$\leftarrow 30$	$\leftarrow 48$
$10 \rightarrow$	4	8	9	10	19
$12 \rightarrow$	3	7	8	9	18
$20 \rightarrow$		3	4	5	14
$32 \rightarrow$					8
$42 \rightarrow$					3

В этой таблице нам нужно выбрать как можно больше клеток так, чтобы в каждой строке и каждом столбце было бы не более одной выбранной клетки, и значения во всех выбранных клетках различались бы. Это можно сделать разными способами, кружками обведён один из возможных способов выбрать 4 клетки. Корабли 10 и 18 будут подбиты выстрелом «4, 14», корабли 12 и 26 подбиты выстрелом «7, 19», корабли 20 и 30 подбиты выстрелом «5, 25», корабли 32 и 48 подбиты выстрелом «8, 40». Оставшиеся два корабля подобьём отдельными выстрелами в другие моменты времени, например, «1, 43» и «2, 26». Итого 6 выстрелов.

Меньшим количеством выстрелов обойтись нельзя, т.к. для этого нужно выбрать в таблице 5 клеток, что невозможно, т.к. корабли 32 и 42 пересекаются каждый только с одним кораблём 48.

Задача 4. Рассади болельщиков

У этой задачи было несколько вариантов, отличающихся файлами с данными. Разберём решение одного возможного варианта.

Сначала посчитаем сумму всех чисел, например, используя формулу =SUM(A1:A1000). Эта сумма равна 54. Значит, в каждой из трёх частей сумма уровней поддержки должна быть равна 18. Нам нужно найти два блока, начинающихся с первой строки. Сумма чисел в первом блоке должна быть равна 18, а размер блока близок к $\frac{1}{3} \cdot 1000$ (это будет первая трибуна). Второй блок — это первая и вторая трибуна вместе, сумма чисел в этом блоке должна быть равна 36, а размер блока близок к $\frac{2}{3} \cdot 1000$.

Для каждой строки посчитаем сумму чисел в блоке до этой строки (включительно). Для этого запишем в ячейку B2 формулу =SUM(A1:A1), затем эту формулу скопируем в блок B1:B1000. Теперь найдём строки, для которых данная сумма равна 18. Это можно сделать при помощи фильтра строк по значению 18 в столбце B.

Найдём две строки в районе строки 333, в которых в столбце B записано число 18. Это строки 315 и 369. Число 315 ближе к $\frac{1}{3} \cdot 1000$. Если выбрать первую трибуну размера 315, то на оставшихся двух трибунах должно быть $1000 - 315 = 685$ болельщиков, то есть границу между второй и третьей трибуной нужно искать в районе строки $315 + 685/2 = 657{,}5$.

Теперь сделаем фильтр по значению 36 в столбце B и посмотрим на найденные строки в окрестности строки 657. Число 36 будет в ячейке B656, то есть есть ответ, в котором сумма размеров первых двух трибун равна 656. Тогда на второй трибуне будет $656 - 315 = 341$ болельщиков, а на третьей — $1000 - 656 = 344$ болельщиков.

Перебором других близких вариантов можно убедиться, что это лучшее решение.

Ответ: 315, 341, 344.

У этого решения разность между наибольшей и наименьшей трибуной равна $344 - 315 = 29$. Это лучшая разность для этого варианта. Решения с большей разностью оценивались в меньшее число баллов. При этом в каждом варианте самое лучшее решение оценивалось в 100 баллов, а решение с правильным, но самым худшим разбиением (с максимальной разностью размеров трибун) оценивалось в 40 баллов. Все остальные решения набирали от 40 до 100 баллов, если сумма уровней поддержки на каждой трибуне была равна 18.

Задача 5. Самолёт

Если $n \leqslant 6$, то ответом будет число 1. Каждая следующая полная или неполная четвёрка кресел добавляет один проход, то есть для $n = 7..10$ ответом будет 2, а для $n = 11...14$ ответом будет 3.

Поэтому для нахождения ответа нужно взять значение $n - 6$ и поделить на 4 с округлением вверх, что можно сделать по формуле $(n - 6 + 4 - 1)//4$, то есть $(n - 3)//4$.

К результату нужно прибавить 1 (один проход для первых 6 рядов), и возьмём максимум из этого числа и 1 (т.к. при маленьких n должен быть хотя бы один проход).

Пример решения на языке Python.

```
n = int(input())
print(max(1, 1 + (n - 3) // 4))
```

Задача 6. Майки и носки

Эта задача оказалась довольно сложной для многих участников, т.к. нужно было разобрать все принципиально возможные случаи. Многие участники набрали по этой задаче от 52 до 58 баллов, рассмотрев только основной случай. Несколько участников при этом решили все остальные задачи на 100 баллов, но так и не нашли полное решение этой задачи.

Первоначально предметно-методическая комиссия считала эту задачу несложной и планировала дать её на школьный этап, однако, в ходе подготовки задачи оказалось, что она куда более интересна, чем предполагалось заранее, поэтому эта задача была отложена на муниципальный этап.

У Саши есть несколько стратегий действий.

1. Выбрать $B+1$ майку и $D+1$ пару носков. Тогда поскольку количество красных маек равно B, то он обязательно вытащит синюю майку. Также Саша обязательно вытащит синюю пару носков, взяв $D + 1$ пару носков. Итого, вытащив $B + 1$ майку и $D + 1$ пару носков, он обязательно получит синий комплект вещей.

2. Аналогично, можно получить комплект из красной майки и красных носков, взяв $A+1$ майку и $C + 1$ пару носков.

Решения, которые рассматривали только два этих случая, набирали 56 баллов.

Самый простой пример, на котором такое решение работает неправильно, это случай $A = B = C = D = 1$. Такое решение выдаст ответ $(2, 2)$, хотя можно вытащить всего три предмета: например, $(2, 1)$ или $(1, 2)$. Это следующие стратегии.

3. Взять $\max(A, B) + 1$ майку. Тогда Саша гарантированно вытащит и синюю, и красную майки и тогда ему достаточно вытащить 1 пару носков, которая может быть любого цвета.

4. Взять 1 майку и $\max(C, D) + 1$ пар носков.

Решение, которое аккуратно разбирает все эти случаи (нужно не забыть, что ещё некоторые из данных чисел могут быть равны 0, поэтому не все указанные случаи возможны), набирает 100 баллов. Пример такого решения.

```
a = int(input())
b = int(input())
c = int(input())
d = int(input())
ans = []
if a > 0 and c > 0:
    ans.append([b + 1, d + 1])
```

```
if b > 0 and d > 0:
    ans.append([a + 1, c + 1])
if a > 0 and b > 0:
    ans.append([max(a, b) + 1, 1])
if c > 0 and d > 0:
    ans.append([1, max(c, d) + 1])
m = min(ans, key=sum)
print(*m)
```

Пояснения по указанному решению. Здесь заводится список `ans`, в который будут складываться возможные варианты стратегии Саши, например, $(b+1, d+1)$ и т.д. Каждая стратегия — это пара чисел (тип данных кортеж). Кортежи мы будем добавлять в список, только проверив, что нет проблем с нулями, например, первая стратегия применима только в случае $a > 0$ и $c > 0$.

Затем мы в списке `ans` находим наименьший элемент используя функцию `min` с параметром `key=sum`. Это означает, что кортежи сравниваются по результату вызова функции `sum` от каждого кортежа, т.е. мы находим кортеж с минимальной суммой элементов.

Также частичные баллы можно набрать при помощи «переборного» решения, рассматривающего все возможные ответы, выбирающее среди них подходящий и наилучший среди подходящих. Например, решение сложности $O((a+b)(c+d))$ можно получить, перебирая количество выбранных маек x от 1 до $a+b$ и количество выбранных пар носков y от 1 до $c+d$. Далее проверим, может ли пара (x, y) являться ответом. Для этого проще проверить, что пара (x, y) **не может быть ответом**. Это происходит, если все выбранные майки и все выбранные носки могут оказаться разноцветными, то есть если $x \leqslant a$, $y \leqslant d$ или $x \leqslant b$, $y \leqslant c$.

В приведённом ниже решении функция `bad(x, y)` проверяет это условие, то есть проверяет, что пара (x, y) **не походит**. Поэтому в основной программе перебираются числа x и y и проверяется условие `not bad(x, y)`.

Пример переборного решения, набирающего 52 балла.

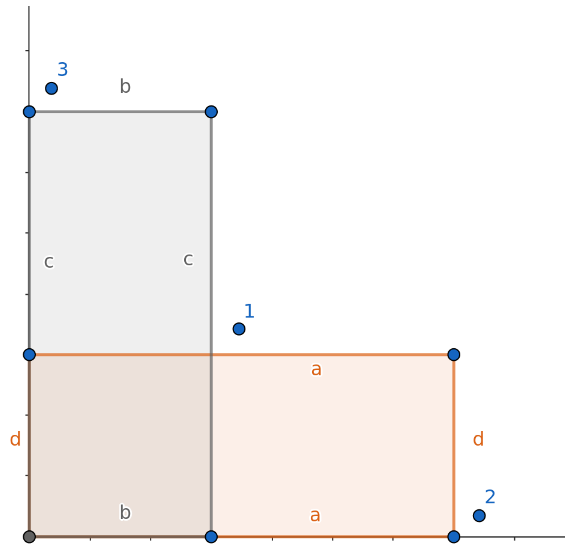
```
a = int(input())
b = int(input())
c = int(input())
d = int(input())
ans1 = a + b
ans2 = c + d

def bad(x, y):
    return x <= a and y <= d or x <= b or y <= c

for x in range(1, a + b + 1):
    for y in range(1, c + d + 1):
        if not bad(x, y) and x + y < ans1 + ans2:
            ans1 = x
            ans2 = y
print(ans1, ans2)
```

Переборное решение на 68 баллов можно получить, если перебирать только одно из чисел в ответе, а второе находить быстро.

Наконец, рассмотрение областей, которые не являются ответом, можно превратить и в решение на полный балл сложности $O(1)$. Пусть (x, y) — рассматриваемый ответ. Точка (x, y) **не подходит**, если она попадает в один из двух прямоугольников: $0 \leqslant x \leqslant a$, $0 \leqslant y \leqslant d$ или $0 \leqslant x \leqslant b$, $0 \leqslant y \leqslant c$. Значит, нужно выбрать точку, которая не принадлежит этим прямоугольникам и имеет минимальное значение $x + y$. В качестве таких точек имеет смысл рассматривать только три точки, изображённые на рисунке. Несложно видеть, что эти точки соответствуют разным возможным стратегиями Саши, описанным в начале разбора.



Задача 7. Произведение цифр

Нам нужно получить минимальное число, произведение цифр которого равно N. В частности, ответ должен содержать как можно меньше цифр. Также отметим, что в ответе не может быть цифры 1 (кроме случая $n = 1$).

В разложение числа N на простые множители могут входить только простые меньшие 10: 2, 3, 5, 7. С делителями 5 и 7 просто, они войдут в ответ, как отдельные цифры. А вот двойки и тройки можно сгруппировать так, чтобы цифр стало меньше. Три двойки можно заменить на 8: $2^3 = 8$. Две тройки можно заменить на 9: $3^2 = 9$. Осталось не более двух двоек и не более одной тройки. Если осталась одна двойка и одна тройка, их можно заменить на 6. Если после этого осталось две двойки, их можно заменить на 4. Если после этого остались двойка или тройка, то они войдут в ответ, как самостоятельные цифры. Полученные цифры нужно вывести в порядке неубывания (от меньшей к большей).

Но есть и более простое решение. Оказывается, «выгодно» выделять как можно большие цифры в числе. Например, если $n = 18$ то из двух вариантов ответа 29 или 36 меньшим будет ответ 29. При этом в ответе 29 есть цифра 9, и если попробовать выделить сначала большую цифру 9, то после деления 18/9 получится меньший результат, который мы сможем использовать для старших разрядов записи ответа.

Получаем «жадное» решение — давайте делить наше число на все возможные цифры от 9 до 2 в порядке уменьшения, то есть попробуем в произведении выделить как можно больше девяток, затем как можно больше восьмёрок и т.д. Если в результате деления от числа n осталась 1, то нужно вывести найденные цифры в обратном порядке. Если же осталось число, большее 1, то значит у исходного n был делитель, больший 10, поэтому нужно вывести «-1». Также нужно отдельно обработать случай, когда $n = 1$, тогда ответ также равен 1. В приведённом ниже решении это происходит, если строка `ans` пустая, то есть если ни одного однозначного делителя у числа n не было найдено.

Пример решения на языке Python.

```
n = int(input())
ans = ""
for d in range(9, 1, -1):
    while n % d == 0:
        n //= d
        ans = str(d) + ans
if n == 1:
    print(ans or "1")
else:
```

```
    print(-1)
```

Решение на 30 баллов можно получить, если перебирать все числа, начиная с 1, пока не найдётся число, произведение цифр которого равно n. Пример такого решения.

```
n = int(input())
ans = 1

def prod(n):
    res = 1
    while n > 0:
        res *= n % 10
        n //= 10
    return res

ans = 1
while prod(ans) != n:
    ans += 1
print(ans)
```