Разбор задач

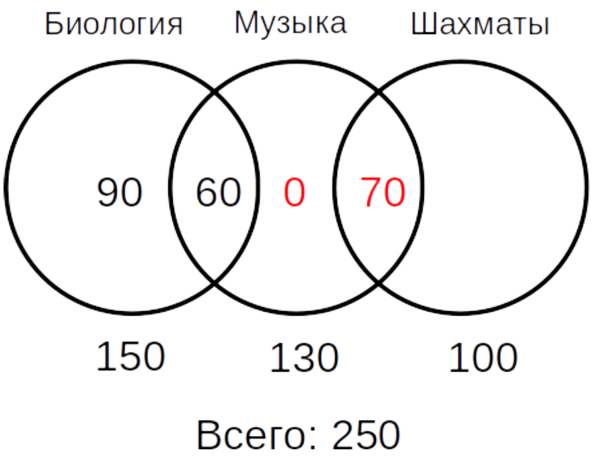
Задача 1. Набор на кружки

1. Так как 150 школьников выбрали биологию, количество учеников в школе не может быть меньше 150. Но оно может быть равно 150, если все ученики будут выбирать биологию и ещё одно или два дополнительных занятия.

2. Наибольшее число учеников в школе окажется в случае, если все выбрали разные занятия. Тогда число учеников будет равно $150 + 130 + 100 = 380$.

3. Если и биологию, и музыку выбрали 85 учащихся, то только биологию выбрали $150 - 85 = 65$ учащихся, только музыку выбрали $130 - 85 = 45$, а ровно один из этих предметов выбрали $65 + 45 = 110$ школьников.

4. Поскольку из 250 учащихся биологию выбрали 100 учащихся, шахматы 150, и никто не выбрал и шахматы, и биологию одновременно, то каждый учащийся обязательно выбрал или биологию, или шахматы, то есть нет учащихся, выбравших только музыку. Каждый учащийся, выбравший музыку, выбрал ещё один предмет. При этом музыку и биологию выбрали 60 учащихся, значит, музыку и шахматы выбрали $130 - 60 = 70$ учащихся.



5. В предыдущем пункте музыку посещают 130 человек, а не посещают $250 - 130 = 120$ человек, при этом только музыку не посещает никто. Чтобы число учеников, посещающих только музыку, стало равным числу учеников, не посещающих музыку, необходимо, чтобы 120 новых школьников записались только на музыку.

Задача 2. Чаепитие

Для чаепития необходима такая расстановка предметов: Х, ..., Х, В, ..., В, Ч, ...Ч. Нам необходимо получить перестановку предметов, для которой получение такой последовательности потребовало бы как можно больше операций. Поэтому в ответе не могут идти буквы Х и В подряд, иначе, переставив их местами, мы получим большее число операций. Также подряд не могут идти буквы В и Ч. То есть ответ всегда имеет вид Ч, ..., Ч, В, ..., В, Х, ..., Х. Осталось только понять, сколько нужно взять чая, варенья и хлеба в ответе.

Пусть в ответе чай встречается x раз, варенье встречается y раз, хлеб встречается z раз, $x + y + z = n$. Посчитаем количество секунд, необходимых для приведения такой перестановки в порядок. Нам придётся поменять местами каждую порцию чая и варенья, это займёт xy секунд. Аналогично понадобится yz секунд, чтобы поменять варенье и хлеб и xz секунд, чтобы поменять чай и хлеб. Нужно подобрать такие значения x, y, z, чтобы сумма $xy + yz + xz$ была максимальной.

Интуитивно понятно, что числа должны быть равны или близки (отличаться на 1). Докажем это. Пусть, например, числа x и y отличаются на 2 и более, то есть $x \geqslant y + 2$. Рассмотрим новую последовательность, в которой x будет на 1 меньше, а y увеличим на 1. Тогда для новой последовательности ответ равен $(x-1)(y+1)+(x-1)z+(y+1)z = xy+x-y-1+xz+yz$, то есть ответ изменится на $x-y-1$, и если $x-y \geqslant 2$, то продолжительность увеличится. Таким образом, в правильном ответе среди чисел x, y, z не должно быть различающихся на 2 и более.

Итак, если n делится на 3, то необходимо взять $x=y=z=n/3$. Если n не делится на 3, то одно или два из этих чисел нужно увеличить на 1, в зависимости от остатка от деления n на 3.

Возможный правильный ответ:

```
ЧВХ
ЧЧЧВВВХХХ
ЧЧЧЧВВВХХХХ
ЧЧЧЧВВВВХХХХХ
```

Задача 3. Кратчайший путь

Можно начать с любого города и выбирать на каждом шаге ещё не посещённый город с минимальной стоимостью перелёта. Получится маршрут «AEDCGFB», при этом в конце этого маршрута будут выбраны уже довольно дорогие перелёты. Стоимость такого маршрута равна 27. Дальше можно начать перебирать различные варианты продолжения, заменяя дешёвые перелёты на более дорогие, в расчёте, что в дальнейшем удастся использовать рейсы меньшей стоимости.

Лучший ответ имеет вид «ABDEFCG», стоимость этого маршрута равна 24.

Задача 4. Путешествие

Добавим в таблицу две строки. В строке 1 будем записывать заголовки столбцов. В строке 2 в ячейках B2 и C2 запишем нули — координаты начальной точки. В последующих 1000 строках столбцов B и C запишем координаты точки, в которой окажется Данис после выполнения очередного шага.

Для этого нам нужно в каждой строке с 3 по 1002 записать формулы в столбцах B и C, учитывающие координаты в предыдущей строке и команду перемещения, записанную в столбце A — к каждой из координат нужно прибавить одно из трёх чисел 0, 1, -1 в зависимости от команды. Например, в ячейку B3 записать формулу **=B2+IFS(A3="направо";1;A3="налево";−1;TRUE();0)**, в ячейку C3 записать формулу **=C2+IFS(A3="вверх";1;A3="вниз";−1;TRUE();0)**, скопировать эти две ячейки в блок B4:C1002.

Ответ на первый вопрос можно найти при помощи формулы или фильтра. Зададим фильтр в столбце B по значению -11 и в столбце C по значению 9. Будет отфильтровано 9 строк, это и есть ответ на первое задание.

Чтобы ответить на второй вопрос, необходимо посчитать количество различных значений в столбцах B и C. Однако, работать с парами чисел трудно, удобно работать с одним числом. Для каждой точки в столбце D запишем уникальное число, которое будет различать координаты. Для этого можно использовать формулу **=B2 * 100 + C2**, которую мы запишем в ячейку D2 и скопируем в блок D3:D1002. Здесь мы воспользуемся тем, что все координаты, как можно заметить, по модулю будут меньше 50.

Далее нужно посчитать количество уникальных чисел в столбце D. В Excel это можно сделать при помощи функции «Удалить дубликаты», в LibreOffice Calc есть параметр фильтра «Без повторений». Мы же рассмотрим решение, не использующее специальные возможности конкретных приложений. Для каждого посещения точки вычислим в столбце E последовательный номер этого посещения (то есть для первого посещения точки запишем 1, при повторном посещении этой точки запишем 2 и т.д.). В ячейку E2 запишем формулу **=COUNTIF(D2:D2;D2)**. Обратите внимание на абсолютную адресацию в формуле: это подсчёт значений, равных D2, среди значений в этом столбце, находящихся выше этой ячейки. Скопируем эту формулу в блок E3:E1002. Тогда при первом заходе в данную точку в соответствующей ячейке таблицы будет записано число 1. Нужно посчитать количество ячеек столбца E, в которых записано число 1. Это можно сделать функцией COUNTIF или при помощи фильтра. Количество таких ячеек будет 383.

Чтобы ответить на третий вопрос, нужно найти точку, которой соответствует максимальное значение в столбце E. Это тоже удобно делать при помощи фильтра. Максимальное значение в столбце E равно 12, отфильтровав строки в столбце E по числу 12, получим единственную строку. Координаты этой точки равны $(-14; 4)$.

Наконец, посчитаем расстояние от каждой точки маршрута до точки $(10; 6)$. Запишем в ячейку F2 формулу **=ABS(B2−10) + ABS(C2−6)** и скопируем её в блок F3:F1002. Снова воспользуемся фильтром, на этот раз по столбцу F. Минимальное значение в столбце F составит 10 и оно будет достигаться в точке $(1; 7)$ (причём эта точка будет посещена дважды). Это и есть ответ на последнее задание.

Задача 5. Качели

Первую группу тестов можно пройти при помощи переборного решения. Будем перебирать значение ответа (массу камня) в переменной ans. Для каждого значения ans проверим, смогут ли дети качаться с камнем данной массы. Переберём все возможные варианты размещения детей и камня, всего таких способов 6 (тремя способами можно выбрать одного ребёнка, который сидит на одном конце качелей, и двумя способами — конец, на который положат камень). Для каждого способа посчитаем модуль разности весов на концах качелей, если он не превосходит d, то ответ найден.

Пример такого решения.

```
a = int(input())
b = int(input())
c = int(input())
d = int(input())
ans = 0
while True:
    if (abs(a+b-c-ans) <= d or abs(a+b-c+ans) <= d
     or abs(b+c-a-ans) <= d or abs(b+c-a+ans) <= d
     or abs(a+c-b-ans) <= d or abs(a+c-b+ans) <= d):
        print(ans)
        break
    ans += 1
```

Чтобы набрать 100 баллов можно в этом решении заменить линейный поиск ответа на двоичный. Но такое решение довольно сложно, т.к. необходимо правильно определить границы для двоичного поиска. Нет нужды приводить такое решение, потому что у задачи есть более элегантное решение сложности $O(1)$.

Для того, чтобы минимизировать разницу весов на концах качелей, необходимо на одну сторону посадить самого тяжёлого ребёнка, а на другую сторону — двух других детей. Посчитаем разницу масс на концах качелей в этом случае, если она не превосходит d, то камень не нужен, и ответом будет 0. Иначе вычтем из этой разницы значение d, это и будет ответ.

Пример такого решения.

```
a = int(input())
b = int(input())
c = int(input())
d = int(input())
side1 = max(a, b, c)
side2 = a + b + c - side1
print(max(0, abs(side1 - side2) - d))
```

Задача 6. Фонари

Вывод программы различается для случаев, когда размещение фонарей возможно или невозможно. Один фонарь первого типа освещает $2x + 1$ домов, второго типа — $2y + 1$ домов. Поэтому сначала проверим, существует ли решение задачи, то есть посчитаем максимальное число домов, которые могут освещаться a фонарями первого вида и b фонарями второго вида. Если это число меньше n, то нужно вывести -1. Иначе получим ответ при помощи «жадного» алгоритма: для минимизации числа фонарей выберем фонарь, который освещает больше домов, и разместим его так, чтобы множество домов, которое он освещает, непосредственно примыкало к уже освещённым домам. Повторим этот процесс, пока все дома не станут освещены.

В приведённом ниже решении мы предполагаем, что фонари первого вида освещают большее число домов, то есть $x \geqslant y$. Если это не так, то поменяем два вида фонарей местами. Поэтому будем стараться всегда использовать фонарь первого вида. В переменной last_lighted хранится номер последнего освещённого дома. Цикл продолжается, пока не все дома освещены, то есть пока last_lighted < n. Если есть ещё фонари первого вида, то используется фонарь первого вида, и количество освещённых домов увеличивается на $2x+1$ для фонарей первого типа и на $2y+1$ для второго типа. При выводе координаты нового освещённого дома необходимо учесть, что координата дома в выводе не может быть больше n.

Пример решения.

```
n = int(input())
a = int(input())
x = int(input())
b = int(input())
y = int(input())

if a * (2 * x + 1) + b * (2 * y + 1) < n:
    print(-1)
else:
    if x < y:
        x, y = y, x
        a, b = b, a
    last_lighted = 0
    while last_lighted < n:
        if a > 0:
            print(min(n, last_lighted + x + 1), x)
            last_lighted += 2 * x + 1
            a-= 1
        else:
            print(min(n, last_lighted + y + 1), y)
            last_lighted += 2 * y + 1
            b -= 1
```

Задача 7. Деление шоколадки

Должен получиться большой кусок и ещё $k-1$ маленьких кусочков, поэтому размер большого куска будет не более, чем $mn-k+1$.

Чтобы набрать 60 баллов можно перебирать размеры большого куска $a \times b$, при этом $1 \leqslant a \leqslant m$, $1 \leqslant b \leqslant n$. Проверим, что $ab \leqslant mn-k+1$ и запомним наибольшее подходящее значение ab.

Такое решение будет иметь сложность $O(mn)$. Пример такого решения.

```
m = int(input())
n = int(input())
k = int(input())
ans = 1
for a in range(1, m + 1):
    for b in range(1, n + 1):
        if a * b <= m * n - k + 1:
            ans = max(ans, a * b)
print(ans)
```

Чтобы набрать 100 баллов, необходимо избавиться от одного из циклов. Заметим, что при фиксированном a значение b, при котором площадь прямоугольного куска будет наибольшей, но не превосходящей $mn-k+1$ можно получить, взяв целую часть от деления $mn-k+1$ на a. Необходимо только учесть, что значение b не может превышать n, поэтому возьмём в качестве наибольшего подходящего b минимум из значений b и (m * n - k + 1) // a. Такое решение будет иметь сложность $O(m)$. Также допустимо перебирать значение длины другой стороны за $O(n)$ или взять наименьшую из двух сторон n или m.

```
m = int(input())
n = int(input())
k = int(input())
ans = 1
for a in range(1, m + 1):
    b = min(n, (m * n - k + 1) // a)
    ans = max(ans, a * b)
print(ans)
```

Мы получили наибольший по площади целочисленный прямоугольник, площадь которого не превосходит $mn-k+1$, помещающийся внутри прямоугольника $m \times n$. Осталось доказать, что такой прямоугольник является ответом на задачу, то есть его и ещё $k-1$ кусков можно получить разламыванием прямоугольника $m \times n$.

Рассмотрим разные значения k. При $k=1$ кусок всего один, его площадь не превосходит mn, ответом является само значение mn и такой прямоугольник мы получим, не делая разломов.

При $k \geqslant 3$ получить большой прямоугольник можно двумя разломами — вдоль каждой из сторон шоколадки. Мы получим нужный прямоугольник и ещё два куска. Если нам необходимо получить больше двух дополнительных кусков, то есть при $k>3$, то станем разламывать меньшие куски на части. Один дополнительный разлом увеличивает число кусков на 1. Куски удастся разламывать до тех пор, пока каждый из них не будет состоять из одной дольки, поэтому всегда можно получить нужное количество частей.

Наконец, при $k=2$ шоколадку нужно разломить на две части, сделав одну из частей как можно больше. Отломим от целой шоколадки полоску $1 \times m$ или $1 \times n$, в зависимости от того, какое из значений m или n меньше. Тогда большой кусок будет иметь размер $(m-1) \times n$ или $m \times (n-1)$. Но именно это и есть максимальный целочисленный прямоугольник, который получится разместить в прямоугольнике $m \times n$, но имеющий меньшую площадь, то есть и в этом случае приведённое решение даст правильный ответ.