# Задача A. Страшные числа

*Автор задачи и разработчик: Сергей Щербаков*

Воспользуемся решетом Эратосфена. Реализация решета Эратосфена за время $\mathcal{O}(n)$ также находит для каждого числа в отрезке минимальный простой делитель. Зная для каждого числа его минимальный простой делитель, можно факторизовать (разложить на простые) любое число $t$ за время $\mathcal{O}(\log t)$. Для этого достаточно просто делить число на минимальное простое в разложении, пока оно не станет равно 1.

Заметим, что максимальное возможное количество простых в разложении числа до $2 \cdot 10^5$ не превосходит 25. Для каждого $k$ от 0 до 25 построим массив префиксного предподсчета, $\texttt{cnt}_{k,l}$ — количество чисел $\leqslant l$, имеющих ровно $k$ простых в разложении. Посчитать такой массив можно с помощью перехода $\texttt{cnt}_{k,l} = \texttt{cnt}_{k,l-1} + (\text{primes}(l) \stackrel{?}{=} k)$.

Теперь на запросы можно отвечать за $\mathcal{O}(1)$: на запрос $(k_i, l_i, r_i)$ достаточно вывести ответ $\texttt{cnt}_{k,r} - \texttt{cnt}_{k,l-1}$.

# Задача B. Кошмар наяву

*Автор задачи: Александр Гордеев, разработчик: Мария Жогова*

Рассмотрим несколько случаев расположения точек относительно круга:

1. Обе точки находятся на круге. Тогда при вращении круга прямая между этими двумя точками будет поворачиваться на тот же угол, значит гарантированно существует такой угол поворота, при котором две интересующие нас прямые будут параллельны.

2. Обе точки находятся вне круга. Тогда при вращении круга вообще не происходит изменения в положении прямых, и достаточно проверить параллельность их исходного положения.

3. Одна точка (не теряя общности, $(x_1, y_1)$) находится на круге, а вторая — вне его. В таком случае прямая может поворачиваться в некотором интервале доступных углов. Заметим, что экстремальные значения угол наклона прямой принимает, когда вектор $\overrightarrow{CQ_1}$ перпендикулярен вектору $\overrightarrow{CQ_2}$, где $C(x, y)$ — центр окружности, а $Q_1$ и $Q_2(x_2, y_2)$ — наши точки.

   Обозначим за $Q_1^1$ и $Q_1^2$ соответствующие этим двум экстремальным углам положения $Q_1$ на повернутом круге. Тогда, чтобы проверить, что прямая $Q_1Q_2$ может быть параллельной второй прямой, достаточно проверить, что направляющий вектор второй прямой лежит между векторами $\overrightarrow{Q_2Q_1^1}$ и $\overrightarrow{Q_2Q_1^2}$.

Все описанные выше проверки можно осуществлять с помощью скалярного и веторного произведений векторов.

Альтернативное решение третьего случая — вместо вычисления $Q_1^1$ и $Q_1^2$ в явном виде (для чего понадобится поворот вектора на угол), можно повернуть первую прямую относительно $Q_2$ так, чтобы она стала параллельна второй (то есть на самом деле параллельно сдвинуть вторую до точки $Q_2$), после чего проверить, что точка $Q_1$ может оказаться на ее пересечении с кругом. Для этого достаточно проверить, что $|CQ_1|$ не меньше расстояния от $C$ до полученной прямой. Все описанные вычисления в таком случае можно совершать в целых числах, не переходя к арифметике чисел с плавающей точкой.

# Задача C. Давайте разделимся!

*Автор и разработчик задачи: Мария Жогова*

Переберем в какое помещение пойдет какой лидер.

Пусть первый лидер имеет безрассудство $a_1$ и пошел к помещение $b_1$, а второй $a_2$ и $b_2$ соответственно.

Пусть в первое помещение пошло $x$ человек,тогда во второе $n - x$, тогда надо минимизировать $\max(a_1 \cdot b_1 \cdot x, a_2 \cdot b_2 \cdot (n - x))$. Если нарисовать графики $a_1 \cdot b_1 \cdot x$, $a_2 \cdot b_2 \cdot (n - x)$ и максимум из этих величин. Нетрудно убедиться, что оптимальный ответ находится в точке пересечения первых двух функций, но ответ должен быть целым, поэтому проверим два соседних целых значения и найдем минимум из них.

# Задача D. Расколбас с Франкенштейном

*Автор задачи и разработчик: Александр Гордеев*

Упорядочим множества чисел в порядке $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$. Рассмотрим два числа $x \in X$ и $y \in Y$, где $X$ и $Y$ — наименьшие из данных множеств, содержащие $x$ и $y$ соответственно.

Заметим, что если, не теряя общности, $X \subset Y$, то $x + y \in Y$ и $x \cdot y \in Y$. Более того, в общем случае ни $x + y$, ни $x \cdot y$ не лежат в каком-то меньшем классе чисел. Например, если $x$ — целое, а $y$ — вещественное, ни их сумма, ни их произведение не обязаны быть целыми или рациональными, в общем случае они будут именно вещественными.

Для $x - y$ верны все те же утверждения, за исключением того, что $x - y$ не обязано быть натуральным, если сами $x, y \in \mathbb{N}$. Например, $3, 5 \in \mathbb{N}$, но $3 - 5 \notin \mathbb{N}$, хотя все еще $\in \mathbb{Z}$.

Таким образом, для решения задачи достаточно было занумеровать классы в указанном порядке от 1 до 4 и сказать, что $\texttt{class}(\mathrm{result}) = \max(\texttt{class}(x), \texttt{class}(y))$. Единственное исключение, которое можно обработать отдельным условием — что ответом на запрос «`N - N`» является «`Z`».

# Задача E. Самая страшная история

*Автор задачи: Тимур Гараев, разработчик: Константин Бац*

Давайте перепишем каждое слово в последовательность пар $\langle w_i, c_i \rangle$, где $w_i$ — номер слова, а $c_i$ — порядковый номер символа в слове. Это делается за один проход по всем символам в каждом слове, то есть за $\mathcal{O}\left(\sum\limits_{i=1}^{n} |s_i|\right)$. Для этого достаточно просто итерироваться по данной во вводе строке $s$, увеличивая $c_i$, а при встрече пробела — обнуляя $c_i$ и увеличивая $w_i$. Альтернативно можно просто считывать слова по одному, используя `cin`.

Далее соединим $n$ получившихся последовательностей пар. Тогда мы получим последовательность пар длины $\sum\limits_{i=1}^{n} |s_i|$, в которой на позиции $i$ записаны $w_i$ и $c_i$, которые равны номеру слова и позиции в слове для $i$-го символа в истории соответственно. Таким образом, ответ для каждой гипотезы — элемент полученной последовательности, который можно получить за время $\mathcal{O}(1)$.

Время работы такого решения — $\mathcal{O}\left(m + \sum\limits_{i=1}^{n} |s_i|\right)$.

# Задача F. Стрелочник

*Автор задачи и разработчик: Владислав Власов*

Заметим, что нам нужно найти кратчайший путь в некотором графе, где ребро может весить 1 или 0. Для решения таких задач подойдет алгоритм Дейкстры или 0-1 bfs.

Теперь поймем, что в каждой клетке может быть выгодно быть в разные моменты, а не только в минимальный возможный. Но при этом если два момента времени $t_1$ и $t_2$ отличаются на время, кратное восьми, то достаточно знать $\min(t_1, t_2)$, так как каждые 8 единиц времени все стрелки возвращаются в исходное положение. Таким образом, вместо обычного $\texttt{dist}[v]$, минимального времени, в которое можно оказаться в вершине $v$, будем хранить восемь массивов $\texttt{dist}_d[v]$ — минимальное время, в которое можно оказаться в вершине $v$, имеющее остаток $d$ по модулю 8.

Пересчет же будет таким же, как и в обычном поиске в ширину, только теперь каждую вершину можно воспринимать как 8 независимых вершин: $(v, d)$ будет соответствовать вершине $v$, в которую мы попали в момент времени $t$, что $t \bmod 8 = d$. Время работы алгоритма равно $\mathcal{O}(n + m)$, но с в 8 раз большей константой.

# Задача G. Уиджа

*Автор задачи: Александр Голубев, разработчик: Даниил Орешников*

Сразу заметим, что если $n = m$, то у второго игрока есть симметричная стратегия: всегда после своего хода оставлять ширину доски, равную высоте. Если первый игрок своим ходом уменьшает $n$ до $n_1 < n$, уменьшим $m$ до $m_1 = n_1$. Поскольку после хода первого игрока всегда $n \neq m$, то у второго гарантированно будет ход.

Случай $n \neq m$ сводится к предыдущему. Но в таком случае выигрывает первый игрок. Первым ходом ему стоит уменьшить наибольшую из двух сторон до равной меньшей, после чего действует симметричная стратегия, описанная выше.

Осталось только выбрать выигрышную сторону, после чего следовать описанной выигрышной стратегии. Решение более общего случая, когда указатель находится не в клетке $(1, 1)$, можно прочитать в разборе усложненной версии олимпиады.

# Задача H. Монстры и люди

*Автор разбора: Даниил Орешников*

Переформулируем задачу: требуется выбрать наибольшее по размеру множество $A$ вершин в графе, чтобы между вершинами этого множества не было ребер. Действительно, это является достаточным и необходимым условием в соответствии с задачей. Скажем, что монстры — это вершины из $A$, люди — вершины из $V \setminus A$, а ребра — информация о том, кто кого обвинил. По условию разрешены все ситуации, кроме ситуации «монстр обвинил монстра», то есть кроме ребра из $A$ в $A$.

Будем решать задачу жадным алгоритмом: пусть существует вершина, в которую не ведет ни одно ребро, в таком случае существует оптимальный ответ, в котором эта вершина является монстром. Если в каком-то ответе рассмотренная вершина $v$ не является монстром, то рассмотрим ребро $v \to u$: либо $u$ — человек, тогда ответ можно увеличить, либо $u$ — монстр, но тогда можно сделать $u$ человеком и $v$ монстром, и ответ не уменьшится.

Таким образом, просто будем поддерживать множество вершин, в которые входит 0 ребер, для каждой из них помечать, что она является монстром, после чего

1. пометим обвиненную ей вершину $u$ человеком;
2. удалим ребро из $u$ в обвиненную ей вершину.

В конце, когда не останется вершин со входящей степенью 0, весь граф разобьется на циклы. Очевидно, что в цикле длины $k$ можно выбрать $\left\lfloor \frac{k}{2} \right\rfloor$ вершин. Все это можно реализовать с помощью dfs с дополнительным флагом.