

Задача А. Участники олимпиад

Автор задачи: Николай Ведерников, разработчик: Константин Бац

Чтобы минимизировать число неучаствовавших, нужно максимально увеличить число учеников, которые участвовали хотя бы в одной олимпиаде. Это достигается, если множества участников трёх олимпиад не пересекаются, то есть каждый участник входит только в одно множество. Тогда общее число участвовавших равно $a + b + c$. Если же сумма превышает n , то можно устроить так, что все n школьников участвуют хотя бы в одной олимпиаде. Поэтому минимальное количество неучаствовавших равно $\max(0, n - (a + b + c))$.

Чтобы максимизировать число неучаствовавших, нужно минимизировать число участвовавших хотя бы в одной олимпиаде. Это возможно, если множества участников максимально пересекаются — например, когда одни и те же ученики входят во все три множества. В этом случае количество участвовавших равно $\max(a, b, c)$. Тогда максимальное число неучаствовавших равно $n - \max(a, b, c)$.

Задача В. Подходящий конверт

Автор задачи: Павел Скобелкин, разработчик: Маргарита Саблина

Заметим, что ширина и высота конверта должны совпадать с высотой и шириной одной из открыток (не обязательно одной и той же) — быть меньше, чем высота и ширина любой из открыток стороны конверта не могут, а если мы сделаем стороны больше, чем стороны открыток, то результат будет не оптимальным. Пусть, если у открытки i не равны высота и ширина, высотой h_i будет считаться меньшая из сторон, а шириной w_i — большая из сторон. Выберем конверт, у которого высота $H_{\max}$ будет большей из всех высот открыток, а ширина $W_{\max}$ будет равна максимальной ширине из открыток. Заметим, что взять конверт с меньшими значениями сторон не получится: взять меньшую ширину нельзя, поскольку из всех значений ширины мы выбрали большую, а любая высота меньше или равна любой ширине. $\forall i: w_i \leq W_{\max}, \forall i: h_i \leq H_{\max}$. Взять конверт с меньшим значением высоты не получится, потому что в него не поместится открытка с наибольшей высотой. Такое решение минимизирует H и W , и, таким образом, минимизирует $H \cdot W$.

Задача С. Объединение амулетов

Автор и разработчик задачи: Егор Юлин

Для решения данной задачи будем пользоваться тем фактом, что $\gcd$ меняется не более $\log A$ раз, где A — максимальное значение из a_i . Тогда найдем префиксные и суффиксные $\gcd$, обозначим их за p_i и s_i соответственно.

Разделим подсчет итоговой суммы $\gcd$ на 3 этапа:

1. Отрезок $[1, n]$, для нахождения ответа необходимо посчитать с помощью нахождения lcm всего массива. Это можно сделать с помощью решета Эратосфена и разложения на простые каждого элемента;
2. Отрезки $[i, j]$, такие что $p_i = p_{i+1}$ и $s_{j-1} = s_j$. Ответом для таких отрезков будет являться $\gcd(p_i, s_j)$, так как lcm точно будет делиться на $\gcd(p_i, s_j)$, а итоговый $\gcd$ не может превосходить этого значения. Такие отрезки можно посчитать быстро, если заранее знать позиции, в которых меняется p_i и s_j ;
3. Остальные отрезки, а именно такие $[i, j]$, что $p_i \neq p_{i+1}$ или $s_{j-1} \neq s_j$, так как $\gcd$ меняется не более $\log A$ раз, то таких отрезков будет не более $O(n \log n)$. И тогда их можно посчитать честным проходом по массиву.

Итоговый ответ будет состоять из суммы всех данных 3-х типов отрезков.

Задача D. Постройка сцены

Автор задачи: Демид Кучеренко, разработчик: Егор Юлин

Для нахождения максимального треугольника, заметим, что одна из вершин всегда будет лежать на границах какой-либо дуги. Это верно, так как если все вершины лежат не на границах дуг, то мы можем "поворачивать" треугольник, пока одна из вершин не окажется на границе.

Тогда пусть у нас зафиксирована вершина, максимальный треугольник, который можно вписать в окружность — равносторонний, если для текущих ограничений его можно вписать, то он будет являться ответом.

Теперь предположим, что ответ дает треугольник, у которого ровно две точки лежат на границе, тогда чтобы максимизировать площадь, третья точка должна лежать на середине дуги, образованной данными точками. Если такой точки нет, то ответом будет треугольник, у которого все три точки лежат на границах.

Покажем, что если у треугольника только одна вершина лежит на границе, то у нас обязательно должен быть равносторонний треугольник. Пусть не так, но тогда мы точно можем подвинуть одну из точек, чтобы площадь нашего треугольника увеличилась (это нельзя сделать только в случае, если треугольник равносторонний).

Тогда ответ находится проверкой 3-х случаев. Когда одна вершина лежит на границах и треугольник равносторонний, когда ровно две вершины лежат на границах, и когда все три вершины лежат на границах.

Задача Е. Кубические кусты

Автор задачи: Николай Ведерников, разработчик: Егор Юлин

Если представить исходный массив как разностный, то есть $b_i = a_{i+1} - a_i$, то тогда в массиве b мы хотим найти самый длинный подотрезок, в котором первая половина состоит из 1, а вторая из -1.

Для нахождения такого максимального подотрезка будем использовать ДО, в котором будем хранить текущий ответ, максимальный префикс для ответа и максимальный суффикс.

Тогда для пересчета мы обновляем ответ, а так же проверяем, можно ли расширить префикс и суффикс.

Изменения высот — это обычные изменения в точке в ДО.

Задача F. Бардак никому не нужен

Автор и разработчик задачи: Павел Скобелкин

Из теоремы Шпрага-Гранди следует, что если посчитать функцию Гранди от каждой нитки, а потом взять xor от всех полученных значений, то мы получим значение $= 0$, если позиция Пети проигрышная, и значение > 0 иначе. Единственная проблема — ограничения на длины ниток в задаче достаточно большие, поэтому функцию Гранди по определению посчитать не получится.

Для решения давайте просто научимся считать функцию Гранди от нитки длины x . Для начала распишем значение функции Гранди по определению:

$$g(x) = \text{mex}\{g(a) \oplus g(b) \mid a + b = x, \gcd(a, b) > 1\}.$$

Утверждается, что:

$$g(x) = \begin{cases} 0, & \text{если } x = 2 \text{ или } x \text{ нечётно,} \\ 1, & \text{если } x \equiv 0 \pmod{4}, \\ 2, & \text{если } x \equiv 2 \pmod{4}. \end{cases}$$

Во время контеста можно было несложно заметить эту закономерность, посчитав функцию Гранди для маленьких длин. Сейчас докажем её по индукции:

База для $1 \leq x \leq 3$ очевидна.

Пусть утверждение верно для всех $s < x$, тогда докажем для x .

Рассмотрим 3 случая:

1. x нечетно. Тогда все разбиения $x = a + b$ имеют вид: (четное число > 2) + нечетное. Четное число не может быть равно двум, так как в таком случае не выполнится второе условие: обязательно будет $\gcd(2, x - 2) = 1$. Значит, по предположению индукции, $g(\text{нечетное}) = 0$, $g(\text{четное}) > 0$, тогда их $\oplus$ будет > 0 , значит, в множестве $\{g(a) \oplus g(b) \mid a + b = x, \gcd(a, b) > 1\}$ все числа будут > 0 , то есть его mex будет равен нулю. То есть, функция Гранди от нечетного числа равна нулю.
2. $x \equiv 0 \pmod{4}$. Возьмем разбиение $x = \frac{x}{2} + \frac{x}{2}$, получим $g(\frac{x}{2}) \oplus g(\frac{x}{2}) = 0$. При этом, $\text{hog} = 1$ получить нельзя, так как разрез на 2 нечетных даст 0, а разрез на 2 четных даст 0 или 3. Значит, mex будет равен 1.
3. $x \equiv 2 \pmod{4}$. Рассмотрев разбиения $n = \frac{n}{2} + \frac{n}{2}$ и $n = 2 + (n - 2)$, получим hog равный 0 и 1 соответственно. Из четности x очевидно, что hog равный 2 получить невозможно. Значит, $g(x) = 2$.

Далее нужно было посчитать функцию Гранди от длин всех ниток и взять hog полученных значений.

Итого, решение работает за $\mathcal{O}(n)$.

Задача Г. Волшебный чемодан

Автор задачи: Михаил Иванов, разработчик: Маргарита Саблина

Самое простое решение этой задачи — сгенерировать квадрат случайным образом. Квадрат представляет собой случайную перестановку чисел от 1 до n^2 . Необходимо, чтобы $2 \cdot n + 2$ сумм были различными. Можно оценить вероятность того, чтобы все такие суммы были различными. Всего различных сумм n чисел от 1 до n^2 $n^3 - n^2 + 1$. Вероятность того, что выбранные из них случайно $2 \cdot n + 2$ числа будут различными, будет стремиться к 1 для больших n . Такая оценка не будет точной, поскольку суммы в квадрате зависят друг от друга, но порядок оценить можно.

Задача Н. Нечетное число подмасок

Автор и разработчик задачи: Ильдар Гайнуллин

Рассмотрим два случая:

- Каждое число встречается в массиве четное число раз.

В этом случае, какое бы число x мы не выбрали, в массиве будет содержаться четное число его подмасок. Таким образом, ответ -1

- Существует число, которое встречается в массиве нечетное число раз.

Пусть x — минимальное из таких чисел. Заметим, что если y — подмаска x , то $y \leq x$. Но все числа $< x$ встречаются четное число раз, а чисел равных x нечетное число. Таким образом, это подходящее решение.

Задача I. Плейлист

Автор и разработчик задачи: Екатерина Ведерникова

Для решения задачи необходимо промоделировать процесс проигрывания плеера с поддержкой очереди песен.

Необходимо хранить текущий момент времени curTime и очередь песен. Для хранения очереди песен удобно использовать структуру дек, так как эта структура позволяет быстро добавлять элементы в начало и в конец.

Чтобы корректно обработать добавления в момент времени $t = 0$, перед началом моделирования процесса нужно поместить все события с $t_j = 0$ в начало очереди.

На каждом шаге происходит одно из двух событий:

- если сейчас играет песня, нужно обработать все события с $t_j \leq curTime + curLen$, добавляя новые песни в начало очереди, затем увеличить $curTime$ на длительность текущей песни;
- если плеер не воспроизводит музыку (очередь пуста), нужно переместить $curTime$ к моменту добавления следующей песни и добавить в очередь все песни с началом в это же время.

После этого нужно выбрать следующую песню из начала очереди и начать её проигрывание. При этом время начала этой песни и её имя записать в ответ.

Каждая песня будет один раз добавлена в очередь и один раз удалена, каждая операция выполняется за $O(1)$. Итоговое время работы $O(n + m)$.

Задача J. Квантовые коты

Автор задачи: Алексей Забашта, разработчики: Алексей Забашта, Андрей Омельченко

Если N — нечётное число, то просто запишем в ячейку i, j число $((i+j-2) \cdot \text{inv2mod}N) \bmod N+1$, где $\text{inv2mod}N = (N+1)/2$ — число, обратное к 2 по умножению по модулю N , то есть $(2 \cdot \text{inv2mod}N) \bmod N = 1$.

Заполнение $(i+j-2) \bmod N+1$ даст корректную расстановку без учёта ограничения на диагональ, так как сложение по модулю N образует группу. Умножение на $\text{inv2mod}N$ сделает корректной диагональ, так как без модулей $(i+i)/2 = i$.

Решим задачу для чётных N . Если $N = 2$, то расстановки не существует, для остальных чётных N — существует.

Представим, что у нас есть $N/2$ вариантов блоков из двух чисел размера 1 на 2, которые можно зеркально отражать. Пронумеруем каждый вариант от 1 до $N/2$ и на i -м варианте блока разместим числа i и $i + N/2$.

Таким блоками нужно замостить квадрат N на N . В высоту будет N блоков, в ширину $N/2$. Рассмотрим шаблон из блоков с подряд идущими номерами от 1 до $N/2$. Заполним таким шаблоном первую и вторую строку. Затем сделаем циклический сдвиг шаблона влево и заполним следующие две строки. И так далее до конца таблицы. Пример полученной таблицы для $N = 6$ можно увидеть на рисунке (а). Далее сделаем циклический сдвиг всей таблицы вверх (см. рисунок (б)).

Затем будем идти по ячейкам таблицы на диагонали, смотреть, к какому блоку они принадлежат, и отражать соответствующий блок, чтобы на диагонали оказалось нужное число. В примере первый блок остаётся неизменным (см. рисунок (в)), а второй отражается (см. рисунок (г)). Результат прохода по диагонали можно увидеть на рисунке (д), не отражённые блоки покрашены зелёным цветом, а отражённые — синим. После этого пройдемся по всем остальным блокам и отразим те блоки, варианты которых уже встречались выше с тем же отражением. Итоговый результат можно увидеть на рисунке (е).

1	4	2	5	3	6
1	4	2	5	3	6
2	5	3	6	1	4
2	5	3	6	1	4
3	6	1	4	2	5
3	6	1	4	2	5

(а)

1	4	2	5	3	6
2	5	3	6	1	4
2	5	3	6	1	4
3	6	1	4	2	5
3	6	1	4	2	5
1	4	2	5	3	6

(б)

1	4	2	5	3	6
2	5	3	6	1	4
2	5	3	6	1	4
3	6	1	4	2	5
3	6	1	4	2	5
1	4	2	5	3	6

(в)

1	4	2	5	3	6
5	2	3	6	1	4
2	5	3	6	1	4
3	6	1	4	2	5
3	6	1	4	2	5
1	4	2	5	3	6

(г)

1	4	2	5	3	6
5	2	3	6	1	4
2	5	3	6	1	4
3	6	1	4	2	5
3	6	1	4	5	2
1	4	2	5	3	6

(д)

1	4	2	5	6	3
5	2	6	3	1	4
2	5	3	6	4	1
3	6	1	4	2	5
6	3	4	1	5	2
4	1	5	2	3	6

(е)

Изначальная расстановка блоков гарантирует, что в каждой строке каждое число от 1 до N встречается ровно один раз, а каждый вариант блока в каждом столбце встречается ровно два раза. Следовательно, отражениями можно всегда добиться, чтобы в каждом столбце каждое число от 1 до N встречалось ровно один раз. Изначальный выбор пар чисел для блоков и расстановка гарантируют, что на диагонали всегда будет стоять блок с нужным числом.

Задача К. Восстановление весов

Автор и разработчик задачи: Ильдар Гайнуллин

Рассмотрим любое ребро (u, v) с весом w , пусть $d_u > d_v$.

- $d_u + w \geq d_v$, потому что иначе есть кратчайший путь от 1 до v с весом $d_u + w$.
- $d_v + w \geq d_u$, потому что иначе есть кратчайший путь от 1 до v с весом $d_u + w$.

Тогда вес каждого ребра должен принадлежать отрезку $[\max(1, d_v - d_u); l]$.

Кроме этого, для каждой вершины $v \neq 1$ должно существовать хотя бы одно такое ребро, что $d_u + w = d_v$.

Заметим, что если все эти условия соблюдены, то расстановка весов является корректной.

Рёбрам, соединяющим две вершины u, v с равными $d_u = d_v$, можно поставить любой вес.

Остальные ребра ориентируем от меньших d_u к большим.

Для каждой вершины расстановка весов рёбер, входящих в неё, является не зависимой от расстановки весов рёбер, входящих в другие вершины.

Таким образом, если $f(v)$ – число способов расставить веса рёбрам, входящим в v , корректно, то ответ равен $f(2) \cdot f(3) \cdot \dots \cdot f(n)$.

Для фиксированной вершины сначала посчитаем число способов расставить веса рёбрам, входящим в эту вершину, чтобы веса лежали в необходимых отрезках. Это значение равно произведению длин отрезков возможных весов ребра по всем рёбрам.

Но в некоторых таких ситуациях реальное расстояние до v будет больше d_v , а именно это в тех случаях, когда не существует ребра (u, v) с весом w , что $d_u + w = d_v$. В этих случаях вес каждого ребра $u \rightarrow v$ строго больше $d_u - d_v$. Поэтому для таких «плохих» случаев количество находится аналогично, и затем его нужно вычесть из текущего значения.

Для рёбер с известными заранее весами достаточно проверить, что их вес лежит в необходимом интервале, в противном случае заменить текущий ответ на 0.

Задача Л. Долгий прыжок домой

Автор и разработчик задачи: Павел Скобелин

Для начала поймём критерий того, что с помощью набора длин прыжков $x_1, x_2, \dots, x_n$ можно прийти из точки 0 до точки L . Из расширенного алгоритма Евклида несложно понять, что из точки 0 линейной комбинацией прыжков мы сможем прийти до точки $g = \gcd(x_1, x_2, \dots, x_n)$. Очевидно, после этого мы сможем посетить все координаты вида $g \cdot k$, где $k \in \mathbb{Z}$.

Далее покажем, что описанные выше точки — все точки, которые возможно посетить с таким набором иксов. Действительно, любой из $x_1, x_2, \dots, x_n$ делится на g , таким образом, координата после прыжка всегда будет кратна g .

После этого мы хотим, чтобы точка L была равна $g \cdot k$ для некоторого целого k . Получается, что необходимое и достаточное условие на набор иксов: $L \div \gcd(x_1, x_2, \dots, x_n)$.

Тогда задача сводится к тому, что нужно найти набор иксов минимальной стоимости, такой, что его $\gcd$ делит L . Значит, от текущего набора нам интересно только его текущий $\gcd$ и стоимость. Заведём ДП:

$dp_{i,z}$ — минимальная стоимость набора, набранного только из первых i акций, который имеет $\gcd = z$. Поймём, как пересчитывать такое ДП: если текущая акция имеет параметры x_i, c_i , то пересчет с предыдущего слоя будет такой:

$$dp_{i,\gcd(w,x_i)} = \min(dp_{i,\gcd(w,x_i)}, dp_{i-1,w} + c_i), \quad 0 \leq w \leq L$$

Таким образом, ДП работает за $\mathcal{O}(n \cdot L \cdot \log C)$. Как найти ответ? Нужно посмотреть все ДП вида $dp_{n,d}$, где d делит $|L|$.

Задача М. Остаток от суммы остатков

Автор задачи: Павел Скобелин, разработчик: Мария Жогова

Заметим, что прямой перебор всех пар (i, j) за $\mathcal{O}(n^2)$ невозможен при таких ограничениях на n . Цель разбора — вывести формулу, позволяющую посчитать сумму за $\mathcal{O}(\sqrt{n})$.

Часть 1. Фиксируем j

Обозначим $T(j) = \sum_{i=1}^n (i \bmod j)$. Тогда $S(n) = \sum_{j=1}^n T(j)$.

Посчитаем $T(j)$.

Рассмотрим числа $1, 2, \dots, n$ и их остатки при делении на j . Они разбиваются на блоки длины j :

$$[1, \dots, j], \quad [j+1, \dots, 2j], \quad \dots$$

и, возможно, последний неполный блок.

Пусть

$$q = \left\lfloor \frac{n}{j} \right\rfloor \quad (\text{сколько полных блоков}), \quad r = n \bmod j = n - qj \quad (\text{длина хвоста}).$$

Тогда:

- Каждый полный блок даёт набор остатков $0, 1, 2, \dots, j-1$, их сумма равна $0 + 1 + \dots + (j-1) = \frac{j(j-1)}{2}$. Таких блоков q , значит вклад полных блоков: $q \cdot \frac{j(j-1)}{2}$.
- Последний (неполный) блок даёт остатки $0, 1, 2, \dots, r$, их сумма равна $0 + 1 + \dots + r = \frac{r(r+1)}{2}$.

Следовательно,

$$T(j) = q \cdot \frac{j(j-1)}{2} + \frac{r(r+1)}{2}, \quad \text{где } q = \left\lfloor \frac{n}{j} \right\rfloor, \quad r = n - qj.$$

Подставим это в сумму:

$$S(n) = \sum_{j=1}^n \left(q \cdot \frac{j(j-1)}{2} + \frac{(n-qj)(n-qj+1)}{2} \right).$$

Для удобства обозначим две части:

$$A = \sum_{j=1}^n q \cdot \frac{j(j-1)}{2}, \quad B = \sum_{j=1}^n \frac{(n-qj)(n-qj+1)}{2}.$$

Тогда $S(n) = A + B$.

Часть 2. Группируем по значениям $q = \left\lfloor \frac{n}{j} \right\rfloor$

Ключевое наблюдение: значение $q = \left\lfloor \frac{n}{j} \right\rfloor$ меняется не при каждом j , а кусочно-постоянно на целых отрезках.

Для фиксированного q множество j , дающих это значение q , образует интервал:

$$j \in [L, R], \text{ где } L = \left\lfloor \frac{n}{q+1} \right\rfloor + 1, \quad R = \left\lfloor \frac{n}{q} \right\rfloor.$$

Эквивалентный способ построения интервалов (удобный в реализации): будем перебирать j слева направо. Пусть сейчас мы стоим в некотором $j = i$. Тогда

$$q = \left\lfloor \frac{n}{i} \right\rfloor, \quad R = \left\lfloor \frac{n}{q} \right\rfloor.$$

Ясно, что для всех j из отрезка $[i, R]$ значение $\left\lfloor \frac{n}{j} \right\rfloor$ совпадает и равно q . После обработки этого отрезка переходим к $i = R + 1$.

Важно, что количество таких отрезков порядка $O(\sqrt{n})$, поскольку после точки $j > \sqrt{n}$ значения q малы и принимают всего примерно $\sqrt{n}$ разных значений, а до $\sqrt{n}$ самих j тоже около $\sqrt{n}$.

Дальше нам нужно уметь быстро посчитать суммы по отрезку $[L, R]$:

$$\sum_{j=L}^R 1, \quad \sum_{j=L}^R j, \quad \sum_{j=L}^R j^2.$$

Обозначим:

$$\text{cnt} = R - L + 1,$$

$$S_1(L, R) = \sum_{j=L}^R j,$$

$$S_2(L, R) = \sum_{j=L}^R j^2.$$

Эти суммы удобно выражаются через префиксные формулы:

$$\sum_{j=1}^x j = \frac{x(x+1)}{2}, \quad \sum_{j=1}^x j^2 = \frac{x(x+1)(2x+1)}{6}.$$

Тогда

$$S_1(L, R) = \frac{R(R+1)}{2} - \frac{(L-1)L}{2},$$
$$S_2(L, R) = \frac{R(R+1)(2R+1)}{6} - \frac{(L-1)L(2L-1)}{6}.$$

Все эти формулы будем считать по модулю M , используя обратные элементы к 2 и 6 (так как M — простое).

Часть 3. Вклад отрезка $[L, R]$ в A

Напомним:

$$A = \sum_{j=1}^n q \cdot \frac{j(j-1)}{2}.$$

На всём рассматриваемом отрезке $[L, R]$ значение q постоянно, то есть

$$A_{[L,R]} = \sum_{j=L}^R q \cdot \frac{j(j-1)}{2} = q \cdot \frac{1}{2} \sum_{j=L}^R (j(j-1)).$$

Заметим, что

$$j(j-1) = j^2 - j.$$

Тогда

$$\sum_{j=L}^R (j(j-1)) = \sum_{j=L}^R j^2 - \sum_{j=L}^R j = S_2(L, R) - S_1(L, R).$$

Отсюда

$$A_{[L,R]} = q \cdot \frac{1}{2} (S_2(L, R) - S_1(L, R)).$$

Часть 4. Вклад отрезка $[L, R]$ в B

Теперь рассмотрим

$$B = \sum_{j=1}^n \frac{(n - qj)(n - qj + 1)}{2}.$$

Обозначим $r = n - qj$. Тогда

$$\frac{r(r+1)}{2} = \frac{(n - qj)(n - qj + 1)}{2}.$$

Раскроем скобки:

$$r(r+1) = (n - qj)(n - qj + 1) = (n - qj)(n + 1 - qj).$$

Перемножим:

$$(n - qj)(n + 1 - qj) = n(n + 1) - qj(2n + 1) + q^2 j^2.$$

Следовательно,

$$\frac{(n - qj)(n - qj + 1)}{2} = \frac{1}{2} (n(n + 1) - q(2n + 1) \cdot j + q^2 \cdot j^2).$$

Тогда вклад отрезка $[L, R]$:

$$B_{[L,R]} = \sum_{j=L}^R \frac{1}{2} (n(n + 1) - q(2n + 1) \cdot j + q^2 \cdot j^2) = \frac{1}{2} [n(n + 1) \cdot \text{cnt} - q(2n + 1) \cdot S_1(L, R) + q^2 \cdot S_2(L, R)],$$

где

$$\text{cnt} = R - L + 1.$$

Часть 5. Итоговая формула

Сложив вклады $A_{[L,R]}$ и $B_{[L,R]}$, получим вклад интервала $[L, R]$ в общий ответ:

$$S_{[L,R]} = A_{[L,R]} + B_{[L,R]}.$$

Остаётся просуммировать это по всем отрезкам, соответствующим различным значениям $q = \left\lfloor \frac{n}{j} \right\rfloor$.

Так как количество таких отрезков $O(\sqrt{n})$, итоговая асимптотика алгоритма будет $O(\sqrt{n})$, что укладывается для $n \leq 10^{12}$.

Часть 6. Реализация по модулю

Вычисления производятся по модулю $M = 998244353$.

Нужно уметь:

- быстро умножать большие числа с взятием по модулю. В C++ для промежуточного произведения можно использовать тип `__int128`;
- считать обратные элементы $\frac{1}{2}$, $\frac{1}{3}$ (а значит и $\frac{1}{6}$) по модулю M через возведение в степень $M - 2$ по малой теореме Ферма;
- аккуратно приводить все формулы S_1 , S_2 , cnt , $A_{[L,R]}$, $B_{[L,R]}$ к арифметике по модулю.

Структура кода:

1. Считать n .
2. Предварительно посчитать $\text{inv}2 = 2^{M-2} \bmod M$, $\text{inv}3 = 3^{M-2} \bmod M$.
3. Определить функции для суммы первых x чисел и суммы первых x квадратов по модулю:

$$\sum_{k=1}^x k = \frac{x(x+1)}{2}, \quad \sum_{k=1}^x k^2 = \frac{x(x+1)(2x+1)}{6}.$$

4. Запустить цикл по отрезкам:

```
i = 1
while i <= n:
    q = n // i
    R = n // q
    L = i
    посчитать вклад от [L, R]
    i = R + 1
```

5. Вывести накопленную сумму по модулю M .

Часть 7. Проверка на примере

Пусть $n = 5$. Тогда вручную:

$i \bmod j$	$j = 1$	2	3	4	5
$i = 1$	0	1	1	1	1
2	0	0	2	2	2
3	0	1	0	3	3
4	0	0	1	0	4
5	0	1	2	1	0

Сумма всех элементов таблицы равна 26. Алгоритм, описанный выше, тоже выдаёт 26.

Итог

Главные идеи решения:

1. Перенумеровать сумму по j и выразить $\sum (i \bmod j)$ для фиксированного j через число полных блоков $q = \lfloor n/j \rfloor$ и хвост.

2. Заметить, что $q = \lfloor n/j \rfloor$ принимает мало разных значений, и обрабатывать целые отрезки $j \in [L, R]$ с одинаковым q .
3. На каждом отрезке выражать вклад через элементарные суммы $1, j, j^2$, которые считаются по формулам.
4. Всё считать по модулю 998244353 и использовать обратные по модулю числа.

Асимптотика решения — $O(\sqrt{n})$, что достаточно для $n \leq 10^{12}$.