Задача A. Тривиально? Или нет?

Так как ответ это количество часов, которые надо потратить на разработку задачи, нужно посчитать количество идей, на реализацию которых нужно время, и сложить с количеством часов, которые надо потратить на не идейную часть задачи. Для этого нужно из общего количества идей вычесть те, которые реализуются мгновенно, то есть $A-C$. И сложить это нужно с B, то есть ответ это

$$A - C + B$$

Задача B. Школьная задача

Большая часть групп позволяла явным образом вычислить НОК всех чисел. Для полного решения требовалось заметить, что 0 подходит независимо от входного массива.

Задача C. Шкафчики

Разбор

Расположение шкафчиков периодично с периодом 3: в каждом блоке цвета идут как белый, синий, белый. Синие шкафчики имеют номера $2, 5, 8, 11, \ldots$, то есть выполняется $i \mod 3 = 2$.

Введём функцию $F(n)$ — количество синих шкафчиков на отрезке $[1, n]$. Синие шкафчики имеют вид $3k-1$. Из условия $3k-1 \leqslant n$ получаем $3k \leqslant n+1$, значит $k \leqslant (n+1)/3$. Следовательно, $F(n) = (n+1)/3$ (берём целую часть).

Ответ на отрезке $[l, r]$: $Ans = F(r) - F(l-1)$. Подставляя формулу, получаем $Ans = (r+1)/3 - l/3$ (в обоих делениях берётся целая часть).

Сложность: $O(1)$ по времени и $O(1)$ по памяти. Для $l, r \leqslant 10^{18}$ используются 64-битные целые числа.

Задача D. Безопасность наше все

В первой подзадаче достаточно было проверить, что требование по сумме характеристик не противоречит требованием на каждую из них по отдельности, и что $|a_i - l_a| + |b_i - l_b| + |c_i - l_c| \leqslant k$

Во второй подзадаче нельзя изменять триплеты, так что нужно проверить, что они подходят в исходном виде, то есть

$$l_a \leqslant a_i \leqslant r_a$$

$$l_b \leqslant b_i \leqslant r_b$$

$$l_c \leqslant c_i \leqslant r_c$$

$$l_s \leqslant a_i + b_i + c_i \leqslant r_s$$

Следующие 3 подзадачи предполагали решение ручным перебором и последующей реализацией его на языке программирования

В 5 и 6 подзадачах можно было перебрать a'_i, b'_i, c'_i – триплет, в который перейдет a_i, b_i, c_i и для него проверить все условия, а дальше проверить, что $|a_i - a'_i| + |b_i - b'_i| + |c_i - c'_i| \leqslant k$

Для получения полного балла нужно было заметить, что прежде всего нужно привести a_i, b_i, c_i к такому виду, что они удовлетворяют ограничениям $l_a, r_a, l_b, r_b, l_c, r_c$, для этого если очередное число из a_i, b_i, c_i больше верхней границы, то его нужно уменьшить, если же оно меньше нижней границы, то его надо увеличить. Потом проверить, что в условия $l_a, r_a, l_b, r_b, l_c, r_c$ вообще возможно получить l_s, r_s, то есть надо проверить, что отрезки $[l_a + l_b + l_c, r_a + r_b + r_c]$ и $[l_s, r_s]$ пересекаются. Дальше можно заметить, что если они пересекаются и расстояние от a'_i, b'_i, c'_i (триплет после изменений под условия $l_a, r_a, l_b, r_b, l_c, r_c$) до их пересечения не больше количества оставшихся изменений, то данный триплет подходит. Данный факт легко проверить и он остается на упражнение читателю.

Задача E. Частичный XOR

Пусть $S = (a_1 + a_2 + \ldots + a_n) \bmod 2$. Тогда заметим, как при выполнении операции меняется выбранный элемент и S. Обозначим так: (a_x, S).

$(0, 0) \Rightarrow (0, 0)$

$(0,1) \Rightarrow (1,0)$
$(1,0) \Rightarrow (1,0)$
$(1,1) \Rightarrow (0,0)$

Видно, что если $S = 1$, то можно сделать ровно 1 операцию, после чего S станет равно 0 и a_x инвертируется. Заметим, что при $S = 0$ применение операции ничего не изменит. Из этого делаем вывод, что решение следующее:

- Если массивы a и b совпадают, то ответ — 0.
- Если массива a и b различаются в одной позиции, то, если $S = 1$, то ответ 1, а иначе -1.
- Иначе ответ -1.

Задача F. Космические приветствия

Приветствие имеет фиксированный вид длины $2k$: $1, k+1, 2, k+2, \ldots, k, 2k$.

Рассмотрим массив a. Нужно найти количество отрезков длины $2k$, которые совпадают с этим шаблоном, и которые полностью лежат в каждом запросе.

Заметим ключевое свойство: если мы нашли одно корректное приветствие, то следующий возможный старт не может начинаться внутри этого отрезка, так как он полностью фиксирован и занимает $2k$ позиций.

Поэтому пересечения невозможны, и можно рассматривать только непересекающиеся вхождения.

Теперь опишем поиск всех вхождений за $O(n)$.

Идём слева направо указателем i. Если подотрезок $a[i \ldots i+2k-1]$ совпадает с шаблоном, то фиксируем его как найденное вхождение и сразу сдвигаем i на $2k$ (то есть перескакиваем весь отрезок). Если не совпадает — увеличиваем i на 1.

Такой жадный проход корректен, так как любые два вхождения не могут пересекаться: внутри одного корректного отрезка невозможно начать другое корректное вхождение.

В результате мы находим все начала приветствий за один проход по массиву, то есть за $O(n)$.

Далее строим массив *good*, где отмечаем найденные начала приветствий.

Построим префиксные суммы $pref$, где $pref[i]$ — количество найденных вхождений на префиксе.

Ответ на запрос $[l, r]$ — это количество таких вхождений, которые полностью лежат внутри отрезка, то есть с начала в позиции i, где $i \in [l, r-2k+1]$.

Если $r - l + 1 < 2k$, ответ равен 0, иначе: $Ans = pref[r-2k+1] - pref[l-1]$.

Сложность решения: $O(n+q)$ по времени и $O(n)$ по памяти.