# Задача A. Красные и синие таблетки

Будем считать, что $r \leqslant b$ (в противном случае мы можем их поменять местами). Заметим, что вы не можете собрать более $r$ пакетов (хотя бы по одной красной таблетке должно быть в каждом пакете), а потому $b$ не может превосходить $r \cdot (d+1)$ (не более $d+1$ синих таблеток в каждом пакете). Таким образом, если $b > r \cdot (d+1)$, то ответ NO. Иначе мы всегда можем сформировать ровно $r$ пакетов.

# Задача B. Установка M ОС

Пусть $cur$ — текущее количество ноутбуков, на которое установлена M ОС (изначально оно равно 1). Тогда за 1 час мы можем увеличить $cur$ на $min(cur, k)$. Отсюда видно, что несколько первых действий значение $cur$ будет удваиваться, а затем, когда станет больше чем $k$, начнет увеличиваться на ровно на $k$.

Процесс, в котором количество ноутбуков удваивается, можно промоделировать с помощью цикла, т. к. количество удвоений не превосходит $logn$. А чтобы посчитать количество необходимых прибавлений $k$, увеличим ответ еще на $\lceil \frac{n-cur}{k} \rceil$.

Обратите внимание, что $\lceil \frac{n-cur}{k} \rceil$ нужно считать без использования вещественных типов данных; чтобы вычислить $\lceil \frac{x}{y} \rceil$ в целых числах, надо целочисленно поделить $x + y - 1$ на $y$ (это сработает, если и $x$, и $y$ неотрицательны, и $y \neq 0$). Если вы будете использовать нецелые числа, это может привести к неправильному ответу из-за погрешности вычислений.

# Задача C. Отчаянная битва

Давайте научимся считать урон для фиксированного значения $k$. Так как эффект яда от $i$-го удара будет действовать $min(k, a_{i+1} - a_i)$ часов для $i < n$ и $k$ часов для $i = n$, то суммарный урон равен $k + \sum_{i=1}^{n-1} min(k, a_{i+1} - a_i)$. Отсюда видно, что чем больше значение $k$, тем больше сумма. А значит мы можем сделать бинарный поиск по $k$ и найти минимальное значение, при котором сумма больше или равна $h$.

# Задача D. Очередная задача про кресты

Попробуем выбрать каждую точку центром креста, ответом будет такая из них, перекрашивание для которой занимает минимальное время. Считать в лоб займет суммарно $O(nm(n + m))$, что слишком медленно для полного решения, однако такое решение набирает 50 баллов. Заметим, что ответ для некоторой клетки $(x, y)$ можно представить как $cnt_{row}[x] + cnt_{column}[y] -$ (1, если $a[x][y]$ белая, иначе 0), где $cnt_{row}[i]$ — это количество белых клеток в строке $i$, а $cnt_{column}[i]$ — то же для столбца $i$. Первые два слагаемых можно посчитать заранее.

Асимптотика решения: $O(nm)$.

# Задача E. Тимофей и кофе

Здесь есть две отдельные задачи:

- Эффективно сгенерировать массив $c$, где $c_i$ - количество рецептов, которые рекомендуют температуру $i$.
- Эффективно ответить на запросы "сколько чисел $c_a, c_{a+1}, c_{a+2}, \ldots, c_b$ имеет значение хотя бы $k$? где $k$ фиксировано во всех запросах.

Существует несколько решений этой задачи с использованием продвинутых структур данных или алгоритмов. Например, концептуально простая идея заключается в следующем: создайте дерево дерево отрезков на массиве $c$. Мы можем рассматривать все рецепты как запросы на обновление диапазона, которые мы можем эффективно выполнять за $O(q \cdot logm)$ (где $m$ - максимум из $a_i$ и $b_i$), используя ленивое распространение.

После всех рецептов мы заменяем все $c_i$ на 1, если оно не менее $k$, и 0 в противном случае. После этого каждый из следующих $q$ запросов представляет собой базовый запрос суммы на отрезке, который может быть выполнен за маленькое время.

Существуют и другие решения: дерево Фенвика с обновлением на отрезке, сортировка событий, sqrt-декомпозиция с бинарным поиском, алгоритм Мо и так далее. Все эти решения подходят, но все они излишни для этой задачи.

Очень простое решение заключается в следующем. Инициализируйте $c$ нулями. Для рецепта, который рекомендует температуру между $l_i$ и $r_i$, мы должны увеличить $c_{l_i}$ и уменьшить $c_{r_i+1}$.

Суммируйте все значения на префиксе, то есть установите $c_i$ равным $c_1 + c_2 + c_3 + \ldots + c_i$. Это можно сделать за один проход по массиву.

Теперь,$c_i$ - это количество рецептов, в которых рекомендуется температура $i$. Если $c_i$ равен по крайней мере $k$, установите его равным 1, в противном случае установите его равным 0.

Постройте массив префиксных сумм $p$ на массиве $c$.

Теперь на каждый запрос, который запрашивает количество допустимых температур между $a$ и $b$, можно ответить просто как $p_b - p_{a-1}$.

Это работает за $O(n + q + m)$, что действительно быстро.