Разбор задачи «Укладка асфальта»

Для начала посчитаем количество перекрестков. Оно равно $(N+1)^2$, так как каждый перекресток образован пересечением вертикальной и горизонтальной дороги.

Теперь нужно посчитать, сколько фрагментов дорог, соединяющих соседние перекрестки нужно покрыть асфальтом. Нетрудно понять, что таких фрагментов должно быть $(N+1)^2-1$. Каждый такой участок имеет длину K. Таким образом, суммарная длина дорог, покрытых асфальтом, равна $K \cdot ((N+1)^2-1)$.

Также нужно не забыть воспользоваться 64-битным типом данных, так как число в ответе получается довольно большим.

Разбор задачи «Очередь в магазине»

Для получение 40 баллов по данной задаче необходимо было проэмулировать процесс, описанный в условии.

Рассмотрим решение, позволяющее набрать полный балл по данной задаче. Понятно, что для того, чтобы получить достаточно быстрое решение, необходимо научиться обрабатывать событие второго типа достаточно быстро. Заведем некоторую переменную d, изначально равную нулю, в которой мы будем накапливать некоторое значение агрессивности, которое нужно прибавить ко всем людям в очереди.

Теперь рассмотрим, как, имея посчитанное значение d, обработать каждое из трех возможных событий. Пусть произошло событие первого типа. Тогда в очередь нужно добавить число $a-d$, так как мы подразумеваем, что число d должно быть прибавлено ко всем элементам очереди. Если произошло второго типа, то нужно увеличить d на y, а первое число очереди увеличить на $x-y$. В случае, если произошло событие третьего типа, нужно извлечь первое число из очереди, и записать в ответ это число, увеличенное на d.

В таком случае все операции обрабатываются за $O(1)$.

Разбор задачи «Драгоценный камень»

Получить 20 баллов можно, просто перебрав всевозможные варианты покупок и продаж турмалина. Существует 2^N последовательностей дней, в которые осуществляются покупки и продажи. Каждый из них можно обработать за $O(N)$. Получаем решение за $O(2^N \cdot N)$.

Рассмотрим решение на 50 баллов. Воспользуемся динамическим программированием. Пусть $dp[i][0]$ — это максимальная прибыли при условии, что мы рассмотрели первые i дней, и при этом в день i произошла покупка. Аналогично, $dp[i][1]$ — максимальная прибыли при условии, что были рассмотрены первые i дней, но в i-й день произошла продажа.

Тогда $dp[i][0] = \max\limits_{j=1}^{i-1} dp[j][1] - a_i$, а $dp[i][1] = \max\limits_{j=1}^{i-1} dp[j][0] + a_i$. Данное решение работает за $O(N^2)$.

Теперь заметим, что можно не считать каждый раз значение максимума, а поддерживать его в некоторой переменной. Будем поддерживать две переменные: max_0 и max_1, в которых будут храниться соответствующие максимальные значения динамики. Тогда описанное выше решение оптимизируется до $O(N)$.

Разбор задачи «Приближение прогрессией»

Для получения 30 баллов можно для каждой арифметической прогрессии считать отклонение за $O(n)$ по формуле, описанной в условии.

Рассмотрим решение для второй группы тестов. В этой группе все арифметические прогрессии являются константными последовательностями, а также массив является константной последовательностью. Это значит, что отклонение каждой пары элементов равно $(a_1-b)^2$. Если просуммировать по всем элементам, получим выражение $n \cdot (a_1-b)^2$.

Рассмотрим решение для третьей группы тестов. Все прогрессии все еще являются константными последовательностями. Рассмотрим формулу для отклонения: $\sum\limits_{i=1}^{n}(a_i-b)^2 = \sum\limits_{i=1}^{n}(a_i)^2 - 2b \cdot \sum\limits_{i=1}^{n} a_i + n \cdot b^2$. Заметим, что числа под знаками суммы никак не зависят от прогрессий. Их можно посчитать один раз и сохранить в качестве значений двух переменных. После этого ответ для каждой прогрессии можно посчитать за $O(1)$.

Рассмотрим решение для четвертой группы: $\sum_{i=1}^{n}(a_1-(b+d\cdot(i-1)))^2=\sum_{i=0}^{n-1}(a_1-(b+d\cdot i))^2=$
$=\sum_{i=0}^{n-1}(a_1)^2-2a_1\cdot\sum_{i=0}^{n-1}(b+d\cdot i)+\sum_{i=0}^{n-1}(b+d\cdot i)^2=n\cdot(a_1)^2-2a_1\cdot\sum_{i=0}^{n-1}(b+d\cdot i)+n\cdot b^2+2bd\cdot\sum_{i=0}^{n-1}i+d^2\cdot\sum_{i=0}^{n-1}(i^2)=$
$=n\cdot(a_1)^2-2a_1\cdot nb-2a_1d\cdot\sum_{i=0}^{n-1}i+n\cdot b^2+2bd\cdot\sum_{i=0}^{n-1}i+d^2\cdot\sum_{i=0}^{n-1}(i^2)=$
$=n\cdot(b-a_1)^2+2d(b-a_1)\cdot\sum_{i=0}^{n-1}i+d^2\cdot\sum_{i=0}^{n-1}(i^2)$. Здесь нужно быстро вычислять две суммы, использующиеся в формуле. Их значение можно вычислить заранее.

Наконец, для того, чтобы решить задачу в общем случае, необходимо раскрыть чуть более сложную сумму. Процесс упрощения выглядит аналогично, поэтому запишем сразу ответ: $\sum_{i=1}^{n}(a_i-(b+d\cdot(i-1)))^2=\sum_{i=1}^{n}(a_i)^2-2b\cdot\sum_{i=1}^{n}a_i-2d\cdot\sum_{i=0}^{n-1}ia_{i+1}+n\cdot b^2+2bd\cdot\sum_{i=0}^{n-1}i+d^2\cdot\sum_{i=0}^{n-1}(i^2)$.

Получаем решение за $O(n+m)$.

Разбор задачи «Последний рубеж»

Для того, чтобы получить 20 баллов, необходимо узнать информацию о каждом замке. Для этого можно спросить информацию о всех замках с номерами от 2 до $N-1$, после чего найти ответ.

Для решения второй подзадачи необходим всего один запрос. Узнаем информацию о замке с номером $N-1$. Если он закрыт, то замок с номером $N-2$ гарантированно открыт, так как всего закрыты ровно два замка. В этом случае, ответ: $N-2, N-1$. В противном случае ответ: $N-1, N$.

Теперь рассмотрим полное решение. Воспользуемся двоичным поиском. Будем поддерживать следующий инвариант: замок с номером L открыт, а замок с номером R закрыт. Изначально это действительно так. Теперь узнаем информацию о замке с номером $M=\lfloor\frac{L+R}{2}\rfloor$. Если замок с номером M открыт, скажем, что левая граница равна M, в противном случае — правая граница равна M.

Будем повторять эти действия до тех пор, пока $R-L>1$. Через некоторое количество итераций, левая и правая границы укажут на соседние замки. Учитывая описанный выше инвариант, замок L открыт, а замок R закрыт. Значит, мы нашли ответ.

Алгоритм сделает ровно $\lceil\log_2(R-L)\rceil$ запросов, что укладывается в ограничения задачи.