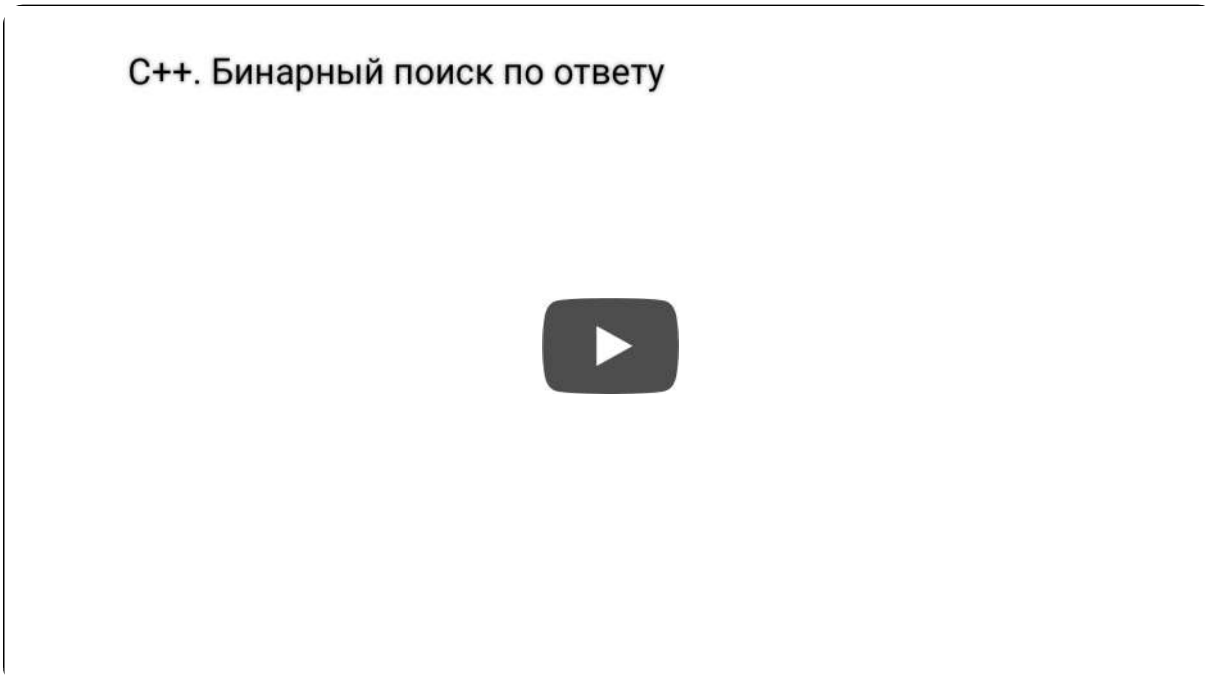
Бинарный поиск по ответу



Бинарный поиск по ответу

Бинарный поиск по ответу позволяет решать определённый ряд задач. Как правило, в этом классе задач величина, которую мы ищем бинарным поиском, и является ответом на задачу.

Рассмотрим общую постановку, которая характерна для задач на бинарный поиск по ответу. Пусть некоторое условие выполнено для всех целых или натуральных чисел не меньших некоторого числа, а для всех чисел меньших этого числа условие не выполнено. Такая ситуация будет иметь место, если из того что условие выполнено для целого числа X, следует, что условие выполнено и для всех целых чисел больших X, а из того что условие не выполнено для целого числа X, следует, что условие не выполнено и для всех целых меньших X (или наоборот). В задаче нас могут попросить найти либо наименьшее целое число, для которого условие выполнено, либо наибольшее целое число, для которого условие не выполнено. Как правило, в задачах, о которых пойдёт речь, поиск необходимого числа X непосредственно (например по формуле) или невозможен, или представляет собой более сложную задачу. Если же для конкретного числа X возможно понять, выполнено для него условие или нет, то есть мы умеем решать в некотором смысле *"обратную задачу"* , то появляется возможность решать изначальную задачу с использованием бинарного поиска по ответу.

Для решения задач подобного вида необходимо выбрать левую границу L бинарного поиска такую, что для неё условие не выполнено, а затем выбрать правую границу R бинарного поиска, для которой условие выполнено. После этого необходимо выполнить некоторое количество итераций алгоритма бинарного поиска. На каждой итерации мы будем находить число $M = \frac{R+L}{2}$, и если для M условие выполнено, то будем сдвигать правую границу $R = M$, а иначе сдвигать левую границу $L = M$. Процесс завершится в тот момент, когда будет выполнено равенство $R = L + 1$, в этот момент число L будет равно максимальному числу, для которого условие не выполнено, а число R – минимальному числу, для которого условие выполнено.

Задача 1. Имеются n дипломов ширины w и высоты h. Необходимо найти минимальное натуральное число X такое, что на квадратном стенде с длиной стороны X, можно разместить все n дипломов без наложения так, чтобы их стороны были параллельны сторонам стенда.

Для решения этой задачи будем использовать тот факт, что при увеличении длины стороны стенда количество дипломов, которое можно разместить на стенде, либо не изменяется, либо увеличивается. Следовательно,есть некоторая длина стенда, которой достаточно для размещения всех дипломов, и любая большая длина будет достаточной для размещения всех дипломов, а любая меньшая длина стороны стенда не будет достаточной для размещения всех дипломов.

Таким образом, необходимую длину X можно искать бинарным поиском. В качестве левой границы бинарного поиска можно выбрать число 0, а в качестве правой границы можно выбрать число $n \cdot \max(w, h)$ или $\lceil \sqrt{n} \rceil \cdot \max(w, h)$.

Теперь, чтобы решить задачу бинарным поиском, необходимо научиться решать ту самую *"обратную задачу"*. Для этого выпишем формулу, по которой для заданной длины X стороны стенда можно найти количество дипломов, которые на этом стенде помещаются. Дипломы выгодно размещать жадно, например заполнить верхнюю часть стенда, пока позволяет ширина, затем заполнить второй ряд дипломов, пока позволяет ширина и так далее, пока хватает высоты стенда. Получится, что по ширине у нас помещаются $\left[\frac{X}{w}\right]$ дипломов, а по высоте — $\left[\frac{X}{h}\right]$ дипломов. Следовательно, на таком стенде помещаются $\left[\frac{X}{w}\right] \cdot \left[\frac{X}{h}\right]$ дипломов, и остаётся сравнить это количество с числом n, чтобы понять, достаточно ли длины X для нашего стенда или нет.

Задача 2. В некоторой игре необходимо убить последовательно n монстров в пещере одного за другим. У монстров есть здоровье $h_0, h_1, \ldots, h_{n-1}$. У героя есть выносливость, которой хватает на m ударов, поэтому необходимо убить всех монстров не более чем за m ударов. Если сила героя равна числу X, то с каждым ударом по монстру у текущего монстра будет убавляться X единиц здоровья, и этот монстр погибнет, когда его здоровье станет меньше или равно нулю. Необходимо найти наименьшее натуральное число X такое, что герой с силой X сможет убить всех монстров не более чем за m ударов, либо определить, что такого X не существует.

Каждого монстра необходимо ударить хотя бы один раз. Поэтому, если $m < n$, то всех монстров убить не получится и искомое число X не существует. Если же $m \geq n$, то решение существует. Если нам хватает силы X, чтобы убить всех монстров не более чем за m ударов, то и большей силы тоже хватит, а если силы X не хватает, то не хватит и меньшей силы, поэтому для этой задачи можно применить бинарный поиск по ответу.

Левую границу бинарного поиска можно выбрать равной нулю $L = 0$, потому что силы ноль нам заведомо не хватит, чтобы убить даже одного монстра. Таким образом, $ok(L) = false$.

Правую границу бинарного поиска можно выбрать равной максимальному здоровью монстра $R = \max(h_0, h_1, \ldots, h_{n-1})$. Действительно, с такой силой мы сможем убивать любого монстра за один удар, и все монстры будут убиты за n ударов, что нам подходит, так как мы рассматриваем тот случай, когда $m \geq n$. Таким образом, $ok(R) = true$.

Теперь, чтобы запустить алгоритм бинарного поиска, остаётся написать функцию $ok(X)$, которая будет возвращать $true$ в том случае, если силы X достаточно, чтобы убить всех монстров не более чем за m ударов, и $false$ в противном случае.

Чтобы с силой удара X убить монстра со здоровьем h_i, необходимо сделать

$\left\lceil \frac{h_i}{X} \right\rceil$ ударов. Поэтому $ok(X) = true$ тогда, когда выполнено неравенство $\left\lceil \frac{h_0}{X} \right\rceil + \left\lceil \frac{h_1}{X} \right\rceil + \cdots + \left\lceil \frac{h_{n-1}}{X} \right\rceil \leq m.$