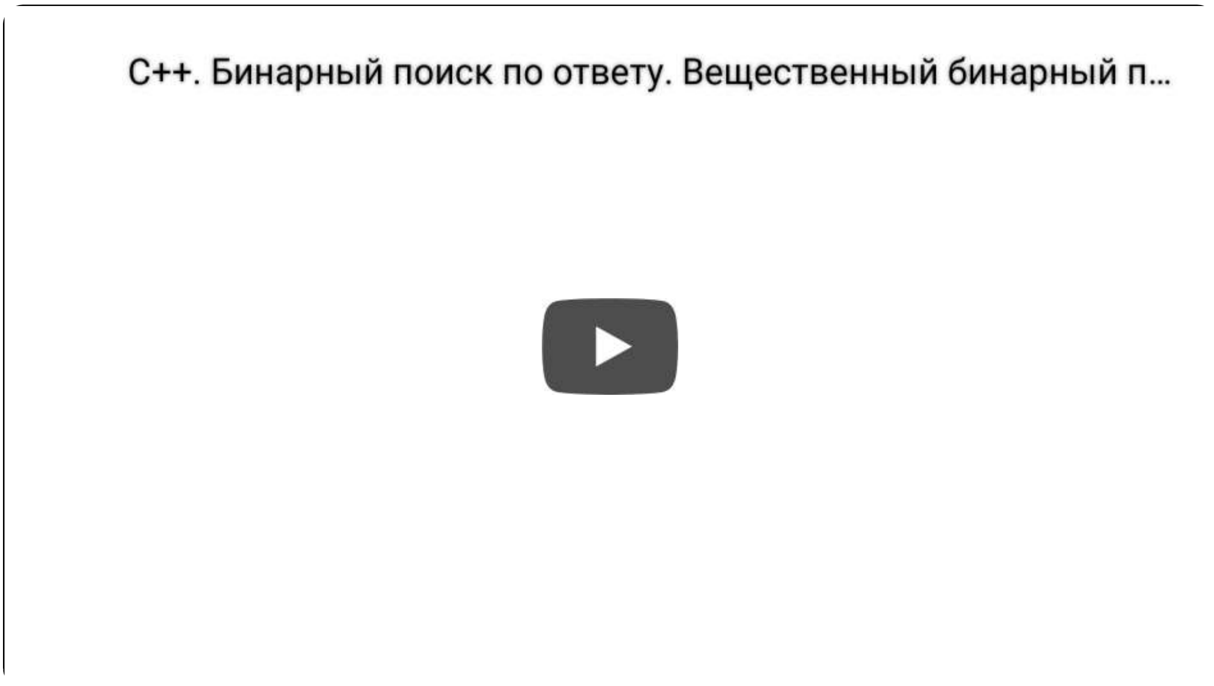
Вещественный бинарный поиск



Вещественный бинарный поиск

Бинарный поиск можно писать не только по целым индексам и значениям, но и для вещественных чисел. Например, некоторая задача на бинарный поиск по ответу может иметь не целое, а вещественное число в качестве ответа. В таком случае практически ничего не меняется, только лишь границы бинпоиска теперь будут вещественными числами и условие остановки алгоритма теперь будет выглядеть иначе.

Если в целочисленном бинарном поиске мы делаем итерации до тех пор, пока $R - L > 1$, то в вещественном бинарном поиске необходимо делать итерации, пока не будет достигнута необходимая точность. Например, можно делать итерации до тех, пока $R - L > \varepsilon$, где ε – достаточно маленькое число. Например, если нужна точность в 6 знаков после запятой, то можно с запасом взять $\varepsilon = 10^{-7}$.

Но есть и другой подход: вместо того чтобы делать итерации, пока границы не станут достаточно близки, можно выполнить фиксированное количество итераций. Так как после каждой итерации алгоритма величина $R - L$ уменьшается в два раза, то при достаточно большом количестве итераций границы автоматически станут близки с большой точностью. Например, если сделать 200 итераций, то этого будет достаточно, чтобы при стартовых границах $R - L = 10^{18}$ в результате окажется, что $R - L < 10^{-18}$. Но тут также стоит учитывать тот факт, что реальные вещественные типы данных имеют ограниченную точность, и это также накладывает ограничение на то, с какой точностью можно вычислить ответ.

Ещё одним отличием вещественного бинарного поиска является то, что в качестве ответа можно выводить любую из двух границ. Потому что это два достаточно близких числа, и каждое из них с некоторой погрешностью является ответом.

Задача 1 Требуется найти число $\sqrt{2}$ с точностью до 4 знака после запятой.

Заметим, что $\sqrt{2}$ можно определить как некоторое положительное число x такое, что $x \cdot x = 2$. При этом выполнено, что если $0 \leq x < \sqrt{2}$, то $x \cdot x < 2$, а если $x > \sqrt{2}$, то $x \cdot x > 2$. Поэтому искомое число можно найти вещественным бинарным поиском. Будем использовать инвариант вида $L \cdot L < 2$ и $R \cdot R \geq 2$, тогда в качестве границ бинарного поиска можно взять $L = 0$ и $R = 2$. Код, решающий эту задачу, может выглядеть следующим образом.

```
double L = 0;
double R = 2;
for (int i = 0; i < 200; ++i) {
    double M = (R + L) / 2;
    (M * M < 2 ? L : R) = M;
}
cout << R << endl;
```

Задача 2 Дана строго возрастающая функция f, определённая на отрезке $[L, R]$ такая, что $f(L) < 0$ и $f(R) \geq 0$. Требуется найти значение $x \in [L, R]$ такое, что $f(x) = 0$.

Условия $f(L) < 0$ и $f(R) \geq 0$ будут инвариантом нашего бинарного поиска. На каждом шаге будем искать середину отрезка $M = \frac{R+L}{2}$, и если $f(M) < 0$, то будем левый конец сдвигать в середину $L = M$, а иначе будем правый конец сдвигать в середину $R = M$. После достаточно большого количества шагов можно будет вывести значение R, которое будет являться ответом с некоторой погрешностью.

Аналогично можно найти решение уравнения $f(x) = C$ для $x \in [L, R]$, где f строго возрастает и выполнено $f(L) < C$ и $f(R) \geq C$. Для этого рассмотрим функцию $g(x) = f(x) - C$, тогда задача сведётся к решению уравнения $g(x) = 0$, которое мы уже разобрали.

Отметим, что задача 1 является частным случаем задачи 2. В задаче 1 мы искали число $\sqrt{2}$, причём искали его как решение уравнения $x^2 = 2$ на отрезке $[0, 2]$. Но это в точности и есть задача 2 с функцией $f(x) = x^2 - 2$, которая является строго возрастающей на отрезке $[0, 2]$.

Задача 3 Треугольник ALR задан координатами своих точек на плоскости. Известно, что для данного треугольника выполнено $|AL| < len$, $|AR| \geq len$, где len — некоторое число, а угол $\angle ALR$ тупой. Требуется найти координаты такой точки X на отрезке LR, для которой выполнено $|AX| = len$.

Вещественный бинарный поиск позволяет решить данную задачу, прибегая к минимальному количеству геометрических формул. Действительно, на каждом шаге алгоритма нам требуется найти координаты точки M — середины отрезка LR, а их легко вычислить как среднее арифметическое координат точек L и R. После этого, зная координаты точек M и A, можно найти длину отрезка AM по теореме Пифагора. Если длина отрезка AM меньше числа len, то можно точку L сдвинуть на место точки M, а в противном случае сдвинуть точку R на место точки M. Выполнив достаточное количество таких итераций (достаточно сделать 200 итераций), мы можем вывести координаты точки R, которые с достаточно хорошей точностью будут являться ответом к задаче.