Алгоритмы и структуры данных. Бинарный поиск

densine

11 октября 2025 г.

Содержание

1. Определение

Стандартная задача: есть загаданное число от 0 до 100 невключительно. Требуется узнать загаданное число за минимальное число вопросов, на которые можно ответить да/нет.

- Чистый перебор ($x == i?$) – $O(n)$.
- Другая идея – делим выборку на два. Т.е. при начальном условии $[l; r]$ получаются отрезки $[l; r//2]$ и $[r//2 + 1; r]$, а затем задаем вопрос: меньше ли данное число, чем $r//2$? В принципе, неважно каким именно будет ответ – интервал сокращается вдвое, и таким образом мы интервал уменшаем до минимального размера (длиной 1), т.е. теперь l, допустим, равно $r//2+1$, а r не изменяется. Аналогично в случае, когда число попадает в меньший интервал. Сложность получается логарифмическая – $O(log_2(n))$. Определим операцию $log_2(n)$:

  $$log_2(n) = k \to min : 2^k \geq x, k \in \mathbb{Z}.$$

  Например, $log(1024) = 10$, а $log(17) = 5$.

Другие полезные их значения:

$$log(10^5) = 17$$

$$log(10^6) = 20$$

$$log(10^8) = 27$$

$$log(10^9) = 30$$

1.1. Примеры задач

- Дано n чисел, есть ли x в нем? Для начала, нам нужно чтобы выполнялось условие:

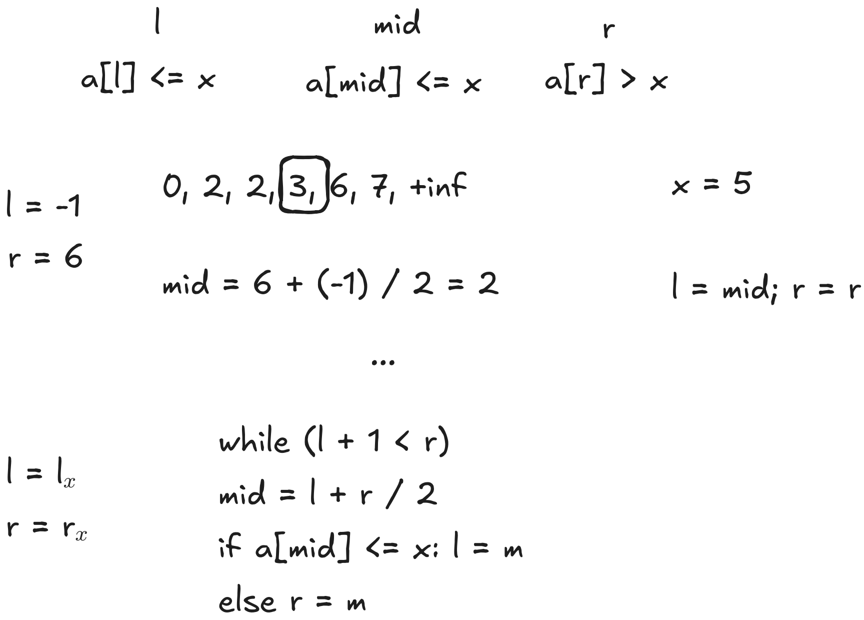
$$a_1 \leq a_2 \leq \cdots \leq a_n$$

```
#include <algorithm>
sort(a.begin(), a.end()); // O(n * log(n))
```

Затем, возможны случаи:

1. $x > a[n/2]$
2. $x < a[n/2]$
3. $x = a[n/2]$

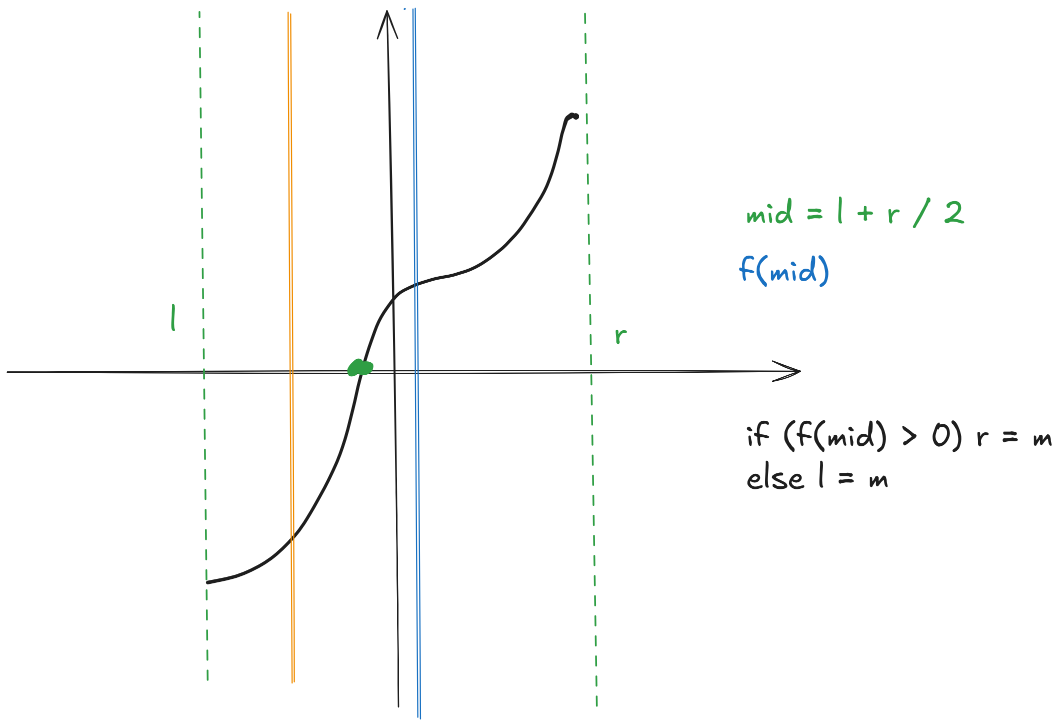
Будем поддерживать $a_l \leq x$ и $a_r > x$. Для начала $l = -1; r = n$. Затем, $mid = l + r/2$.



В итоге ответом будет l – если есть такое число массиве, иначе будет указывать на максимально число, идущее до. Аналогично, r – будет указывать на минимальное число после x.

2. Примеры/реализация

Пусть $f(x) = x^3 + 5$. Она возрастающая и монотонная. Задача – найти $f(x) = 0$.



Еще есть непокрытые аспекты:

1. Изначальные l, r
2. Когда остановиться?

Можно поставить любой $f(l) < 0$, $f(r) > 0$, допустим $l = -1e18$ и $r = 1e18$. С моментом, когда остановиться, тоже легко – в задачах обычно указывается допустимая погрешность, допустим $eps = 10^{-6}$. Тогда итерируемся, пока $r - l > eps$. Или же можно просто повторить ~ 100 итераций – этого с запасом хватит для чисел порядка 10^60.

2.1. Бинарный поиск по ответу

Есть два принтера, первый печатает страницу за x секунд, второй – за y. Принтеры можно использовать параллельно. Найти минимальное время, за которое можно напечатать N страниц. Допустим, ответ T. Тогда за это время мы распечатали $T/X + T/Y$ страниц на принтерах. Нам нужно, чтобы это число было больше или равно N. $f(T) = (T/X + T/Y) - N$. Необходимо найти $f(T) \geq 0$. Просто ищем до тех пор, пока $r - l > 1$.

3. Встроенные функции

Существует несколько функций, которые запрещено использовать в контесте, но никто не запрещает вам использовать их на олимпиаде:

```
lower_bound(a.begin(), a.end(), x); // итератор, указывающий на наименьший
                                    // после найденного
upper_bound(a.begin(), a.end(), x); // итератор, указывающий на наименьший
                                    // после данного и не равный ему
```