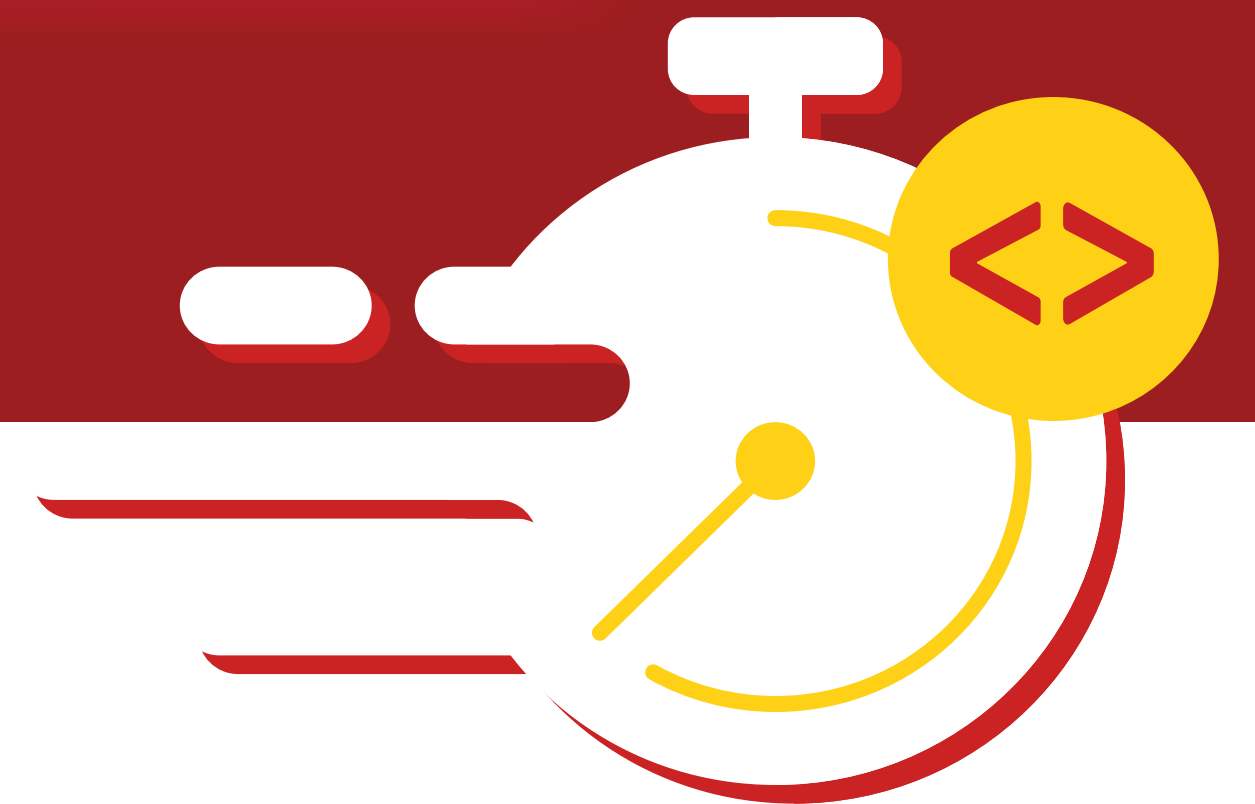
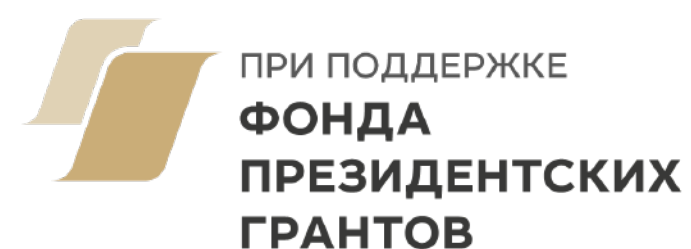


Бинарный поиск: поиск элемента в массиве

Урок 1.3.2



На этом уроке_

- Рассмотрим более жизненную задачу поиска элемента в отсортированном массиве



Задача о поиске элементов в массиве_

- В функцию передается (без копирования) массив $a = a[0..n-1]$, состоящий из чисел; также передается число x
- Известно, что $-10000000000 \leq a[0] \leq a[1] \leq \dots \leq a[n-1] \leq 10000000000$, а $n \leq 100000$
- Требуется вернуть `true`, если x присутствует в массиве, и `false` иначе



Возможные модификации задач_

lower_bound: найти позицию первого элемента, не меньшего x

- Решение: вывести число r

upper_bound: найти позицию первого элемента, строго большего x

- Решение: то же, но с инвариантом $a[l] \leq x < a[r]$

```
1  #include <iostream>
2
3  using namespace std;
4
5  const int MAXN = 100500;
6  int n;
7  int a[MAXN];
8
9  bool isNumberIn(int x) {
10     int l = -1;
11     int r = n;
12
13     // инвариант: a[l] < x <= a[r]
14     // считаем, что a[-1] = -INF,
15     // a[n] = INF
16     while (l + 1 < r) {
17         int mid = (l + r) / 2;
18         if (a[mid] < x)
19             l = mid;
20         else
21             r = mid;
22     }
23
24     return (r < n && a[r] == x);
25 }
```


Итоги_

- Научились искать элемент в отсортированном массиве за $O(\log n)$, а также модификации: `lower_bound`, `upper_bound`

