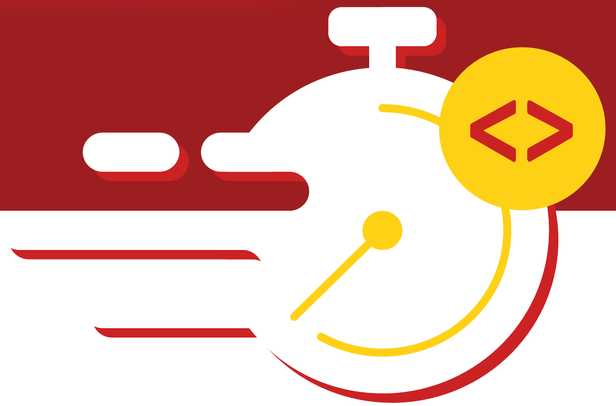
Бинарный поиск

Урок 1.3



МФТИ
МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

ПРИ ПОДДЕРЖКЕ
ФОНДА
ПРЕЗИДЕНТСКИХ
ГРАНТОВ

RUCODE
festival

Фонд развития
Физтех-школ

На этом уроке_

- Метод, который работает за более быструю асимптотику – O(log n)

Задача о нулях и единицах_

- В функцию передается (без копирования) массив a = a[0..n-1] из нулей и единиц
- Известно, что существует число k, 0 <= k <= n, т.ч. первые k элементов массива a суть нули, а все остальные — единицы
- Само k нам неизвестно и требуется найти

Задача о нулях и единицах_

- Наивный метод решения задачи будет работать за $\Theta(k)$, что в худшем случае есть $O(n)$.
  Можно ли быстрее?

Задача о нулях и единицах: примерх_

Рассмотрим следующий пример.
Пусть n = 15, а k нам неизвестно. Тогда:

- a[7] = ? Пусть оказалось a[7] = 0
- a = [0,0,0,0,0,0,0,0,?,?,?,?,?,?,?]
- a[11] = ? Пусть a[11] = 1
- a = [0,0,0,0,0,0,0,0,?,?,?,1,1,1,1]
- a[9] = ? Пусть a[9] = 1
- a = [0,0,0,0,0,0,0,0,?,1,1,1,1,1,1]
- a[8] = ? Пусть a[8] = 0
- a = [0,0,0,0,0,0,0,0,0,1,1,1,1,1,1], и k = 9

Задача о нулях и единицах: обобщение метода_

Пусть дан массив a длины n

Рассмотрим элемент a[mid], mid = (n-1)/2

Возможны два случая:

- a[mid] = 0; тогда mid + 1 <= k <= n;
- a[mid] = 1; тогда 0 <= k <= mid.

Задача о нулях и единицах: обобщение метода_

- В обоих случаях: диапазон уменьшился примерно в два раза!
- Повторяя операцию нужное число раз, и получим искомое k

Задача о нулях и единицах: конкретизация метода_

- Пусть l, r – тот самый диапазон, в котором лежит число k. Более точно, мы будем поддерживать инвариант, что a[l] = 0, a[r] = 1
- Изначально, l = -1, r = n; для удобства будем мысленно считать, что a[-1] = 0, a[n] = 1
- Запускаем цикл while; цель каждой итерации – уменьшить r-l в два раза

Задача о нулях и единицах: время работы_

- Выполнено x итераций. Так как каждая итерация уменьшает r-l в два раза, то после x итераций r-l должно уменьшиться в 2^x раз.
- Необходимо, чтобы длина исходного отрезка была хотя бы 2^x, т.е. $2^x <= n+1$
- Количество итераций цикла, равно как и общая сложность работы алгоритма, O(log n)

```
#include <iostream>

using namespace std;

const int MAXN = 100500;
int n;
int a[MAXN];

int getFirstOne() {
    int l = -1;
    int r = n;
    while (l + 1 < r) {
        int mid = (l + r) / 2;
        if (a[mid] == 0)
            l = mid;
        else
            r = mid;
    }
    return r;
}
```