

Алгоритмы поиска

Лекция 7

ДВОИЧНЫЙ ПОИСК

- Массив должен быть отсортирован

```
def bin_search(l, r, x)
    if l == r - 1
        return (a[l] == x)
    m = (l + r) / 2
    if x == a[m]
        return True
    if x < a[m]
        return bin_search(l, m, x)
    else
        return bin_search(m, r, x)
```

Верхняя и нижняя граница

- Нижняя граница — такое наименьшее i , что $a[i] \geq x$
- Верхняя граница — такое наименьшее i , что $a[i] > x$

Верхняя граница

```
upper_bound(l, r, x)
    if l == r - 1
        return l
    m = (l + r) / 2
    if x < a[m]
        return upper_bound(l, m + 1, x)
    else
        return upper_bound(m + 1, r, x)
```

Нижняя граница

```
lower_bound(l, r, x)
    if l == r - 1
        return l
    m = (l + r) / 2
    if x <= a[m]
        return lower_bound(l, m + 1, x)
    else
        return lower_bound(m + 1, r, x)
```

Вещественный двоичный поиск

- Функция монотонная
- $f(l) \leq 0$ и $f(r) \geq 0$

```
def bin_search(l, r)
    while r - l > EPS
        m = (l + r) / 2
        if f(m) < 0
            l = m
        else
            r = m
    return r
```

Двоичный поиск по ответу

Пример задачи:

На прямой, в конкретных точках расположены стойла (в конкретных точках, n штук) Необходимо расставить K коров так, чтобы минимальное расстояние между коровами было как можно больше

Троичный поиск

- Унимодальная функция

```
ternary_search(l, r)
    for i = 0 to ITN
        m1 = l + (r - l) / 3
        m2 = r - (r - l) / 3
        if f(m1) < f(m2)
            r = m2
        else
            l = m1
    return r
```