Лекция по АиСД

Разбор задачи о минимуме в очереди. Бинарный поиск.

Бинарный поиск

Пусть у нас есть отсортированный массив $a1, a2, ..., a_n$, $a_i < a_{i+1}$.

Мы хотим найти в нем число x

Тривиальное решение

Пишем цикл

```
for elem in a:
    if elem == x:
        return True
return False
```

Сложность – O(n), хотим быстрее

Бинарный поиск

Рассмотрим некоторый элемент массива a_i и сравним его с x
Вспомним, что массив отсортирован
1. Если $a_i < x$, то и все $a_j < x, j < i$ (Левее i нашего элемента точно нет)
2. Если $a_i > x$, то и все $a_j > x, j > i$ (Правее i нашего элемента точно нет)
3. Если $a_i = x$, то мы нашли наш элемент

Бинарный поиск

$a_i < x$

$a_i > x$

Бинарный поиск

Пусть l, r – границы отрезка, в котором мы сейчас ищем элемент

Изначально $l = 0$, $r =$ len(a), то есть ищем во всем массиве

Рассмотрим середину текущего отрезка $a[m], m = (l + r)//2$ 1. Если $a_m < x$, то продолжим поиск только в правой половине массива (l = m)
2. Если $a_m > x$, то продолжим поиск только в левой половине массива (r = m)
3. Если $a_m = x$, то мы нашли наш элемент

Остановимся, когда l + 1 = r (когда мы сошлись к отрезку из одного элемента)

Сложность – O(log len(a))

Пример

$x = 7$

0	2	3	5	7	11	18
0	1	2	3	4	5	6

$l = 0$ $r = 7$

Пример

$X = 7$

0	2	3	$\boxed{5}$	7	11	18
0	1	2	3	4	5	6

$l = 0$ m $r = 7$

$$m = \frac{0+7}{2} = 3$$

$$\underset{5}{a[m]} < \underset{7}{x} \Rightarrow l = m$$

Пример

$X = 7$

~~0~~	~~2~~	~~3~~	5	7	11	18
0	1	2	3	4	5	6

$l = 3$

$r = 7$

Пример

$x = 7$

~~0~~	~~2~~	~~3~~	5	7	$\boxed{11}$	18
~~0~~	~~1~~	~~2~~	3	4	5	6

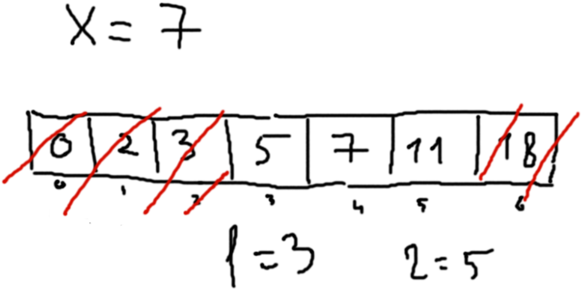
$l = 3$ $\qquad r = 7$

$m = \frac{3+7}{2} = 5$

$a[m] > x \Rightarrow r = m$

11 $\qquad$ 7

Пример



Пример

$X = 7$

~~0~~	~~2~~	~~3~~	5	7	11	~~18~~
0	1	2	3	4	5	6

$l = 3 \qquad r = 5$

$$m = \frac{l + r}{2} = \frac{3 + 5}{2} = 4$$

$a[m] = X!$

Бинарный поиск: код

```
def binary_search(a, k):
    l = 0
    r = len(a)

    while (r - l > 1):
        m = l + (r - l) // 2
        if a[m] > k:
            r = m
        else:
            l = m
    return a[l] == k
```