

# Спортивное программирование. Бинарный поиск.

Плотников Даниил Михайлович, Закарлюка Иван Владимирович

Санкт-Петербургский Государственный Университет

# Оглавление

- 1 Определение
- 2 Бинпоиск по функции
- 3 Бинпоиск по ответу

Задача: существует отсортированный массив

`array = [1, 2, 3, 6, 7, 8, 9, 10, 11, 12]`

Нужно узнать входит ли в массив число  $x$  и если входит то какой индекс имеет

Тривиальный способ — проверить каждый элемент на равенство с  $x$

Тривиальный способ — проверить каждый элемент на равенство с  $x$   
Сложность —  $O(n)$

array = [1, 2, 3, 6, 7, 8, 9, 10, 13, 15]

x = 9

Идея — если  $x > a[i]$ , то индекс  $x > i$   
если  $x < a[i]$ , то индекс  $x < i$

Если выбирать элемент посередине, то массив будет делиться на 2 части



Если выбирать элемент посередине, то массив будет делиться на 2 части  
Размер подотрезка где может быть элемент после  $k$  шагов  $= \frac{n}{2^k}$

Если выбирать элемент посередине, то массив будет делиться на 2 части

Размер подотрезка где может быть элемент после  $k$  шагов  $= \frac{n}{2^k}$

Нам нужно чтобы такой подотрезок был длины 1  $\Rightarrow$  шагов нужно сделать  $\log n$

Значит сложность поиска —  $O(\log n)$

```
1 int l = 0, r = n
2 while (l + 1 < r) {
3     int m = (l + r) / 2;
4     if (a[m] > x) {
5         r = m;
6     } else {
7         l = m;
8     }
9 }
10 return l;
11
```

## Другой взгляд

Бинпоиском можно искать значение нестрого монотонной функции

Массив — в целом тоже функция  $\text{array}(i)$ , возвращающая элемент под индексом  $i$

Обычно бинпоиск на самом деле ищет первое большее/меньшее/больше либо равное/меньше либо равное

## Другой взгляд

Бинпоиском можно искать значение нестрого монотонной функции

Массив — в целом тоже функция  $\text{array}(i)$ , возвращающая элемент под индексом  $i$

## Пример задачи

Есть  $n$  прямоугольников одинакового размера:  $w$  в ширину и  $h$  в длину. Требуется найти квадрат минимального размера, в который можно упаковать данные прямоугольники. Прямоугольники при этом нельзя поворачивать.

Придумать как это решить математически или конструктивно — достаточно сложно

Придумать как это решить математически или конструктивно — достаточно сложно

Однако проверить влезут ли они в квадрат со стороной  $x$  — легко



Придумать как это решить математически или конструктивно — достаточно сложно

Однако проверить влезут ли они в квадрат со стороной  $x$  — легко

Решение — бинпоиск по стороне квадрата

# Код решения

```
1 long long l = 0, r = 1e18;  
2 while (l + 1 < r) {  
3     long long m = (l + r) / 2;  
4     long long s = (m / w) * (m / h);  
5     if (s > n) {  
6         r = m;  
7     } else l = m;  
8 }  
9 return l;  
10
```

Нередко встречаются задачи сводимые к подобным:

- 1 Ваня хочет получить не меньше  $x$  рублей...найдите минимальный  $x$ , что его удовлетворяет
- 2 Даня хочет получить не больше  $x$  пересдач...найдите максимальный  $x$ , что его удовлетворяет

Проблема в том, что обычно условия сложнее

Давайте представим первую как функцию  $f(x)$

- 1  $f(x) = 1$ , если Ваня доволен( $x$  удовлетворяет условиям задачи)
- 2  $f(x) = 0$ , если Ваня недоволен( $x$  не удовлетворяет условиям задачи)

Тогда массив  $[f(1), f(2) \dots f(\text{много})]$  выглядит как  $[0, 0, \dots, 1, 1, 1, \dots]$  — функция монотонна!

## Пример задачи

На прямой расположены  $n$  столбов, в которые необходимо расставить коров так, чтобы минимальное расстояние между коровами было как можно больше. У каждого столба есть координата, на которой оно расположено. Нужно найти минимальное расстояние между коровами при самой оптимальной расстановке.

## По какому параметру делать бинпоиск?

# По какому параметру делать бинпоиск?

## По какому параметру делать бинпоиск?

Поскольку он у нас один свободный, очевидно что по минимальному расстоянию между коровами



# Какую функцию задать?

## Какую функцию задать?

$f(x) = 1$ , если есть расстановка, что нет коров ближе чем на  $x$ , 0 иначе

## Какую функцию задать?

$f(x) = 1$ , если есть расстановка, что нет коров ближе чем на  $x$ , 0 иначе  
Монотонная функция, подходит. Теперь ищем наибольший  $x$ , что  $f(x) = 1$

# Решение

```
1 int l = 0, r = 1e9;
2 while (l + 1 < r) {
3     int m = (l + r) / 2;
4     int remain = k - 1;
5     int prev = st[0];
6     for (int i = 1; i < n; i++) {
7         if (st[i] - prev >= m) {
8             prev = st[i];
9             remain--;
10        }
11    }
12    if (remain > 0) {
13        r = m;
14    } else l = m;
15 }
16 return l;
17
```