НОД — наибольший общий делитель

Пример: $НОД\,(12;\,16) = 4$

max a : $12 \vdots a$ $16 \vdots a$

$12 \rightarrow$ 1 2 3 **4** 6 12

$16 \rightarrow$ 1 2 **4** 8 16

$$НОД\,(a,\,b) \overset{b>a}{=} НОД\,(a,\,b-a)$$

P $a \vdots p$; $b \vdots p$ $a \vdots p_1$; $(b-a) \vdots p_1$

max P max p_1 $b \vdots p_1$ и $a \vdots p_1$

$$\Rightarrow \quad p = p_1$$

$НОД\,(12,\,16) = НОД\,(12,\,16-12) = НОД\,(12,\,4) =$

$= НОД\,(12-4;\,4) = НОД\,(8,4) = НОД\,(4,4) = НОД\,(4,0)$

4

$$\text{НОД}(3;7) = \text{НОД}(3;4) = \text{НОД}(3;1) =$$
$$= \text{НОД}(2;1) = \text{НОД}(1;1) = \text{НОД}(0;1) = 1$$

Ⓘ Алоритм (Евклида) — Вычитание

a — число
b — число
НОД = ?

пока $a \mathrel{!}= 0$ и $b \mathrel{!}= 0$
{

    если $a > b$
      ↳ $a = a - b$

    иначе
      ↳ $b = b - a$

}

если $a == 0$ ↳ Выведи b

иначе
    ⟶ Выведи a

Ⅱ   НОД   (Евклида)   — по остатку от деления

$16 = 12 \cdot 1 + 4$
$16 = 4 \cdot 4 + 0$

24 и 36

$36 = 24 \cdot 1 + 12$: P

: P   : P

остаток : P

НОД $(24; 36)$ = НОД $(24, 36 \% 24)$

P : $24 : P$
       $36 : P$

Алгоритм:

пока $a ! = 0$ и $b ! = 0$
{
        если $b > a$:
                ↳ $b = b \% a$ // остаток от деления
        иначе
                ↳ $a = a \% b$
}
        если $a == 0$
                ↳ вывести b

13   и   11

13 % 11 → 2

НОД (13 ; 11) = НОД (2 ; 11) = НОД (2 ; 1) =

= НОД (0 ; 1) = 1

НОК – наименьшее общее кратное

min   НОК (a, в) = p , что   p : a   и   p : в

НОК (3, 9) = 9 , 9 : 9 ~ 9 : 3

НОК (4, 5) = 20 , 20 : 4 ~ 20 : 5

НОК (4, 10) = 20 ; 20 : 4 ~ 20 : 10

$a = a_1 \cdot a_2 \cdot a_3 \cdot \ldots \cdot a_к$

$в = в_1 \cdot в_2 \cdot \ldots \cdot в_m$

$p = НОК(a, в) = \dfrac{a_1 \, a_2 \, \ldots \, a_к \cdot в_1 \cdot \ldots \cdot в_m}{a_2 \ a_7 \ в_4 \ldots в_9} =$

$$= \text{НОК}(a,b) = \frac{a \cdot b}{\text{НОД}(a,b)}$$

Ⅲ) Разложение на простые множители:

$20 = 2 \cdot 2 \cdot 5$

```
68 | 2
34 | 2
17 | 17
 1 | 1
```

divider = 2

пока N != 1

{

    если N : divider

        ↳ вывеedи (divider)

          N = N / divider

    иначе

        ↳ divider ++

}

$N : 4 \Rightarrow N : 2$

Бинарный поиск

1) Поиск элемента в массиве

$$0 \quad 1 \quad 2 \quad \cdots \cdots \quad (N-1)$$

arr =

elem → есть ли elem в arr
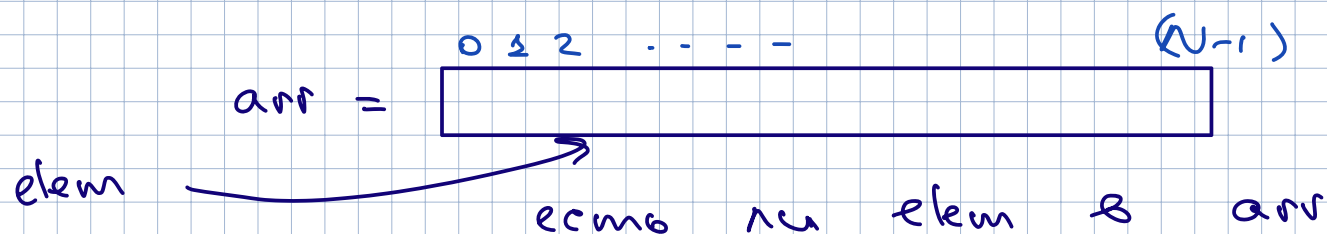
Наивный алгоритм:

```
id = 0
пока    id < N
{
        если  arr[id] == elem
              ↳ вывести (нашли)
              ↳ break
        иначе :
              ↳ id ++
}
если     id == N
         ↳ вывести ( такого элемента
                     нет )
```
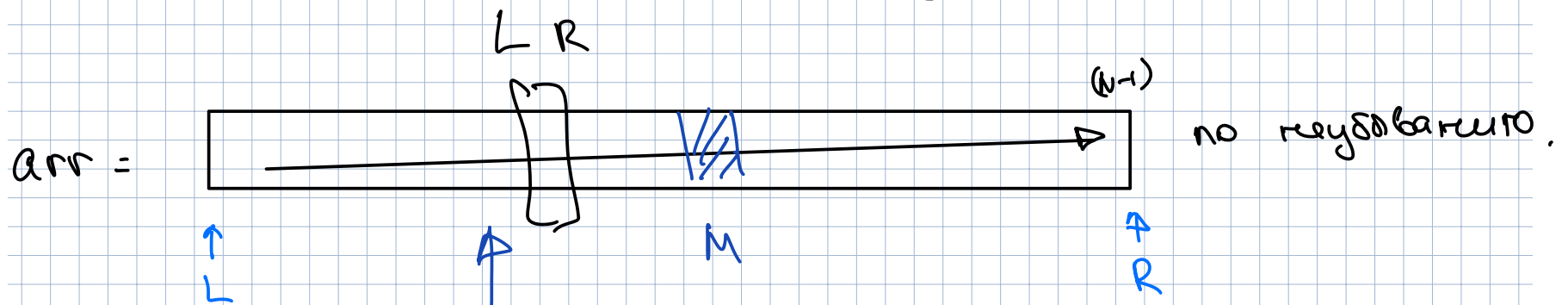
Асимптотика : $O(N)$

2) Бинарный поиск.
работает только на упорядоченных
данных.

arr =

по требованию.

$elem < arr[M] \Rightarrow R = M$

$elem > arr[M] \Rightarrow L = M$

arr = { ... }   // отсортирован
                        по требованию.

L = 0

R = (N-1)

flag = False

пока   L < R - 1

{

M = $\frac{L+R}{2}$

```
если     arr[M] > elem:
              R = M
иначе если      arr[M] < elem:
              L = M
иначе:
         flag = True
         break
}
если     [flag == True] or [arr[L] == elem] or
                          [arr[R] == elem]
         ↳ Выведи (нашли)
иначе
         ↳ Выведи (не нашли)

Асимптотика:  O(logN)
```
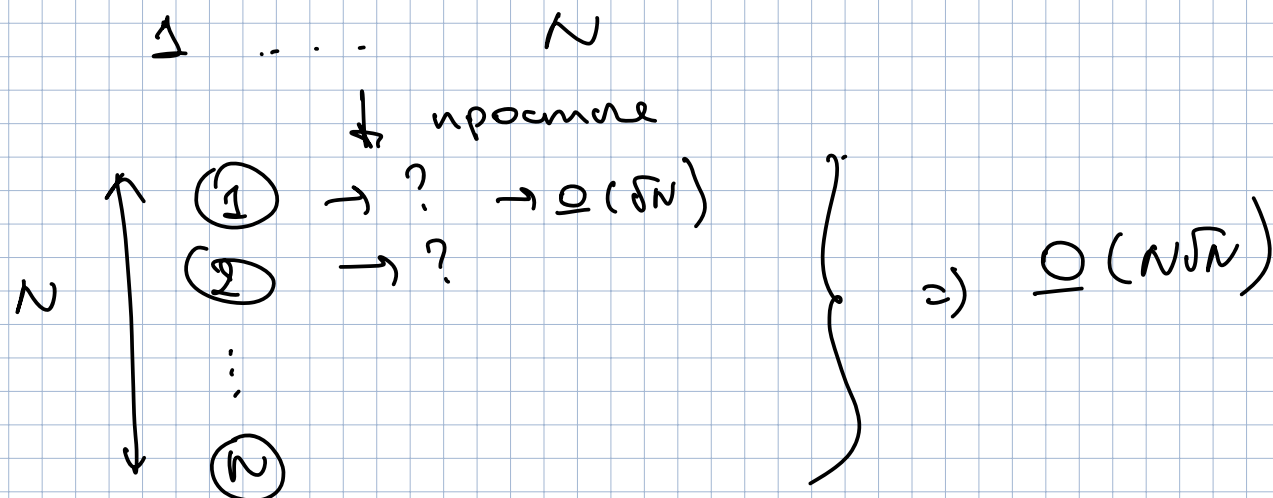
(N) Решето Эратосфена

Нахождение простых чисел ≤ N

1 .... N

↓ простое

① → ? → $\underline{O(\sqrt{N})}$

② → ?

⋮

Ⓝ

N

$\Bigg\}$ =) $\underline{O(N\sqrt{N})}$

0 1 2 . . . . N

primes= | 0 0 0 . . . . 0 |

1 - составное
0 - простое

primes[2] = 0   // простое

Перебор i = 2 ..... N

если primes[i] == 0

$k = 2$

пока $k \cdot i <= N$:

    primes $\{ki\} = 1$

    $k{+}{+}$

$$O(n \log\log n)$$

Перебор $i = 2 \dots N$:

    если primes $\{i\} == 0$:

        Вывели (i)

0	1	2	3	4	5	6	7	8	9	10
				✗		✗		✗	✗	✗

2 3 5 7