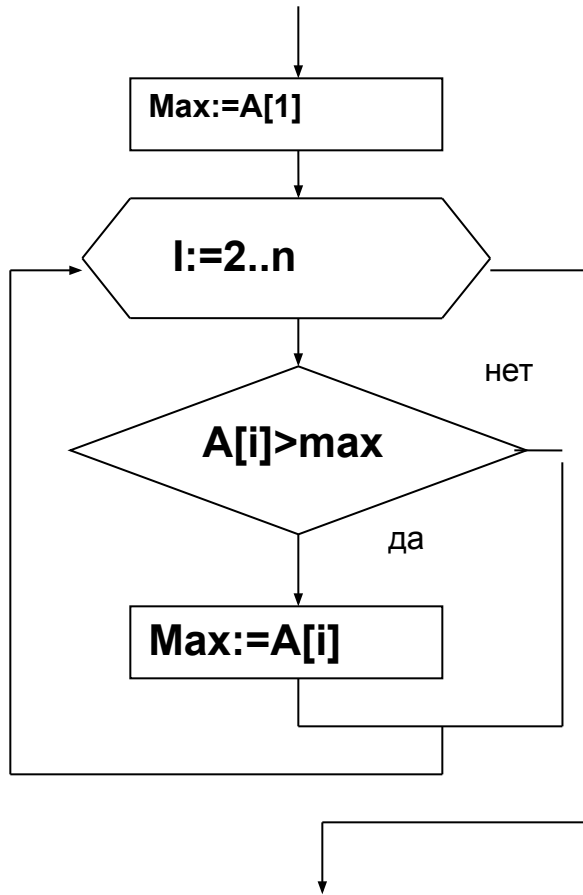


# Оценка сложности алгоритмов

На примере обработки  
массивов

Тютина Т.В.  
учитель информатики и ИКТ  
МБОУ СОШ № 95

# Алгоритм нахождения наибольшего элемента в массиве



Количество операций сравнения на 1 меньше, чем количество элементов в массиве, т. е.  $N-1$

Сложность алгоритма обозначается  $T(N)$

$$T(N) = N - 1$$

# Алгоритм поиска элемента в массиве размерности N

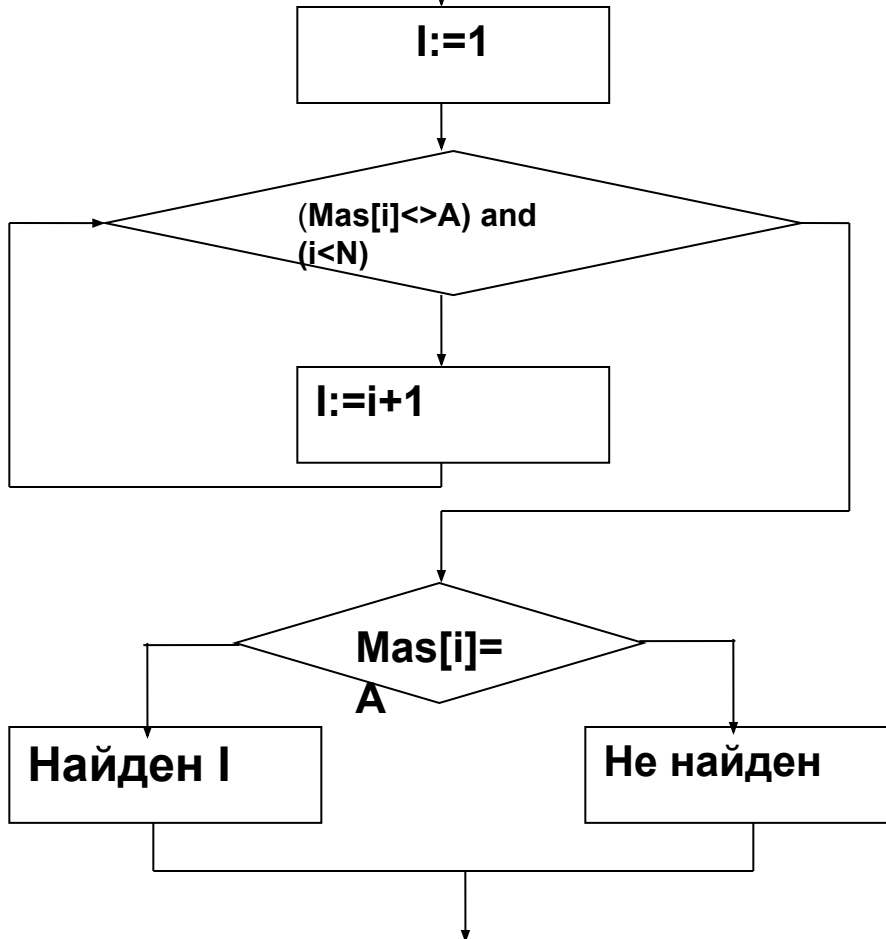
Mas:array[1..N] of integer;

A- некоторое искомое значение

**Идея алгоритма:** нужно двигаться по массиву и сравнивать каждую ячейку с заданным значением. Как только будет обнаружено равенство либо достигнут конец массива, то необходимо остановиться и выдать сообщение.

# Алгоритм поиска элемента в массиве размерности N

$$T(N)=N$$



# Алгоритм поиска в упорядоченном массиве размерности $N$

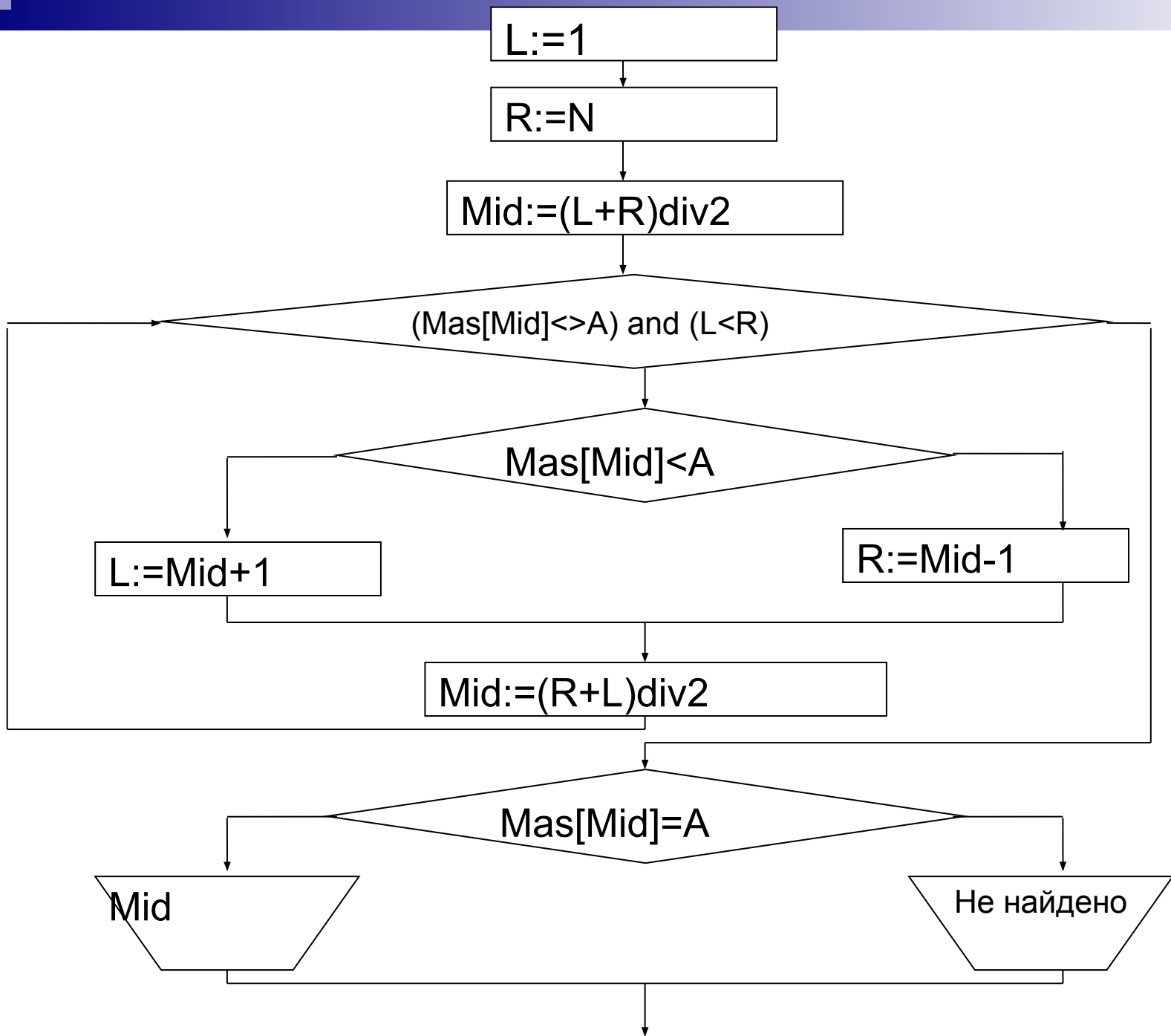
Идея алгоритма: остановиться на первом элементе, большем того, который ищем, т.е. в условии заменить

$$\text{Mas}[i] < A$$

Сложность алгоритма  $T(N)=N$

# Алгоритм поиска в упорядоченном массиве размерности $N$

- Идея алгоритма:
1. Возьмём элемент, стоящий в середине массива. Если он равен  $A$ , то алгоритм закончен. Если элемент  $> A$ , то искомый элемент находится в левой половине массива, а правую половину можно больше не рассматривать.
  2. Аналогично, если элемент  $< A$ , то искомый элемент в правой половине.
  3. С оставшейся частью массива выполняем аналогичные действия.
  4. Эти действия повторяем пока нужный элемент не будет найден или в массиве не останется элементов.




Количество шагов  $n$  (сложность алгоритма) и размер массива  $N$  связаны формулой:

$$2^n = N$$

$$T(N) = \log_2 N$$





**Задача:** упорядочить массив (другим массивом пользоваться нельзя)

**Способ решения:** «пузырьковая сортировка»

**Идея решения:** массив можно упорядочить, меняя местами некоторые элементы, стоящие в «неправильном порядке».

**Ограничимся сравнением и обменом только соседних элементов в массиве и начнём сравнение с конца.**

# Ход сортировки

1.) исходный массив

3 7 9 4 1 5 2 8

не меняем местами 2 и 8

3 7 9 4 1 5 2 8

меняем местами 5 и 2

3 7 9 4 1 2 5 8

не меняем 1 и 2

3 7 9 4 1 2 5 8

меняем местами 4 и 1

3 7 9 1 4 2 5 8

меняем местами 9 и 1

3 7 1 9 4 2 5 8

меняем местами 7 и 1

3 1 7 9 4 2 5 8

меняем местами 3 и 1

~~3~~ 1 7 9 4 2 5 8

Сделав один проход по массиву, мы поставили на место наименьший элемент

# Ход сортировки

2.) Повторяем проход с конца массива, но теперь не доходя до первого элемента.

1 3 7 9 4 2 5 8

не меняем местами 5 и 8

1 3 7 9 4 2 5 8

не меняем местами 5 и 2

1 3 7 9 4 2 5 8

не меняем 4 и 2

1 3 7 9 2 4 5 8

меняем местами 2 и 9

1 3 7 2 9 4 5 8

меняем местами 2 и 7

1 3 2 7 9 4 5 8

меняем местами 2 и 3

1 2 3 7 9 4 5 8

Сделав второй проход, поставили на место второй элемент.

# Ход сортировки

3.) Сделав  $N-1$  проход по массиву –  
упорядочим весь массив.

# Программа

Один проход по массиву:

```
For j:=N downto 2 do
  if Mas[j-1]>Mas[j] then
    begin
      Tmp:=Mas[j];
      Mas[j]:=Mas[j-1];
      Mas[j-1]:=Tmp
    end;
End;
```

# Программа

Этот проход надо повторить  $N-1$  раз.

```
For i:=2 to N do
```

```
  For j:=N downto i do
```

```
    if Mas[j-1]>Mas[j] then
```

```
      begin
```

```
        Tmp:=Mas[j];
```

```
        Mas[j]:=Mas[j-1];
```

```
        Mas[j-1]:=Tmp
```

```
      End;
```

# Сложность алгоритма

Алгоритм делает порядка

$(N-1)+(N-2)+\dots+2+1=N(N-1)/2$  шагов, что примерно равно  $N^2$ .

$$T(N) = N^2$$

При  $N=1000000$  элементов потребуется примерно  $1000000^2=10^{12}$  шагов ( $10^3$ с)