

# Семинар 9

Введение в программирование на Python

Папулин С.Ю.  
papulin\_hse@mail.ru

2015

1. Работа с файлами
2. Решение задач


1. Методы определения min/max
2. Обмен значений переменных
3. Методы сортировки

# Определение min/max


# Определение min/max

Найти минимальное значение в списке

```
myList = [8, 5, 6, 7, 4, 10, 3, 1, 2]
```

1      `val = min(myList)`  1


2      **def** getMinValue(lNumbs):  
        locVal = lNumbs[0]  
        **for** i **in** range(1, len(lNumbs)):  
            **if** (lNumbs[i] < locVal):  
                locVal = lNumbs[i]  
        **return** locVal

`val = getMinValue(myList)`  1


# Определение min/max

Найти максимальное значение в списке

```
myList = [8, 5, 6, 7, 4, 10, 3, 1, 2]
```

1      `val = max(myList)`  10

2      **def** getMaxValue(lNumbs):  
        locVal = lNumbs[0]  
        **for** i **in** range(1, len(lNumbs)):  
            **if** (lNumbs[i] > locVal):  
                locVal = lNumbs[i]  
        **return** locVal


`val = getMaxValue(myList)`  10

# Определение min/max

Найти индекс первого минимального значения в списке

```
myList = [8, 5, 6, 7, 4, 10, 3, 1, 2]
```

```
def getMinIndx(lNumbs):  
    locIndx = 0  
    for i in range(1, len(lNumbs)):  
        if(lNumbs[i] < lNumbs[locIndx]):  
            locIndx = i  
    return locIndx
```


```
indx = getMinIndx(myList)       7
```

# Определение min/max

Найти индекс первого максимального значения в списке

```
myList = [8, 5, 6, 7, 4, 10, 3, 1, 2]
```

```
def getMaxIndx(lNumbs):  
    locIndx = 0  
    for i in range(1, len(lNumbs)):  
        if(lNumbs[i] > lNumbs[locIndx]):  
            locIndx = i  
    return locIndx
```

```
indx = getMaxIndx(myList)       5
```



# Определение min/max

Найти значение и индекс первого минимального элемента в списке

```
myList = [8, 5, 6, 7, 4, 10, 3, 1, 2]
```

```
def getMinIndxValue(lNumbs):  
    locIndx = 0  
    for i in range(1, len(lNumbs)):  
        if (lNumbs[i] < lNumbs[locIndx]):  
            locIndx = i  
    return (locIndx, lNumbs[locIndx])
```

```
indx, val = getMinIndxValue(myList)
```



indx: 7  
val: 1

# Определение min/max

Найти значение и индекс первого максимального элемента в списке

```
myList = [8, 5, 6, 7, 4, 10, 3, 1, 2]
```

```
def getMaxIndxValue(lNumbs):  
    locIndx = 0  
    for i in range(1, len(lNumbs)):  
        if(lNumbs[i] > lNumbs[locIndx]):  
            locIndx = i  
    return (locIndx, lNumbs[locIndx])
```

```
indx, val = getMaxIndxValue(myList)
```




indx: 5  
val: 10

Если в списке несколько значений, соответствующих максимальным/минимальным значениям, как вывести все индексы этих элементов?

# Обмен значений переменных

`a = 3; b = 10`

① `b, a = a, b`  `a = 10; b = 3`

② `tmp = a`  
`a = b`  
`b = tmp`  `a = 10; b = 3`

# Сортировка пузырьком

# Сортировка пузырьком

Входной список/массив:

8	5	6	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

**Проход 1:**

Сравнение

↓	↓							
8	5	6	7	4	10	3	1	2

Обмен

5	8	6	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

Сравнение

	↓	↓						
5	8	6	7	4	10	3	1	2

Обмен

5	6	8	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

# Сортировка пузырьком

5	6	8	7	4	10	3	1	2
5	6	7	8	4	10	3	1	2

5	6	7	8	4	10	3	1	2
5	6	7	4	8	10	3	1	2

5	6	7	4	8	10	3	1	2
---	---	---	---	---	----	---	---	---

5	6	7	4	8	10	3	1	2
5	6	7	4	8	3	10	1	2

5	6	7	4	8	3	10	1	2
5	6	7	4	8	3	1	10	2

5	6	7	4	8	3	1	10	2
5	6	7	4	8	3	1	2	10

Проход 2:

5	6	7	4	8	3	1	2	10
---	---	---	---	---	---	---	---	----

5	6	7	4	8	3	1	2	10
---	---	---	---	---	---	---	---	----

5	6	7	4	8	3	1	2	10
---	---	---	---	---	---	---	---	----

5	6	4	7	8	3	1	2	10
---	---	---	---	---	---	---	---	----

5	6	4	7	8	3	1	2	10
---	---	---	---	---	---	---	---	----

5	6	4	7	8	3	1	2	10
---	---	---	---	---	---	---	---	----

5	6	4	7	3	8	1	2	10
---	---	---	---	---	---	---	---	----

5	6	4	7	3	8	1	2	10
---	---	---	---	---	---	---	---	----

5	6	4	7	3	1	8	2	10
---	---	---	---	---	---	---	---	----

5	6	4	7	3	1	8	2	10
5	6	4	7	3	1	2	8	10

Max  
проход 1

Max  
проход 2

# Сортировка пузырьком

Проход 3:

5	6	4	7	3	1	2	8	10
---	---	---	---	---	---	---	---	----

⋮

5	4	6	3	1	2	7	8	10
---	---	---	---	---	---	---	---	----

Проход 4:

5	4	6	3	1	2	7	8	10
---	---	---	---	---	---	---	---	----

⋮

4	5	3	1	2	6	7	8	10
---	---	---	---	---	---	---	---	----

Проход 8:



1	2	3	4	5	6	7	8	10
---	---	---	---	---	---	---	---	----



# Сложность сортировки пузырьком

Сложность	Наилучший случай	Средний случай	Наихудший случай
Вычисление	$O(n)$	$O(n^2)$	$O(n^2)$
Память	$O(1)$	$O(1)$	$O(1)$

# Сортировка выбором

# Сортировка выбором

Входной список/массив:

8	5	6	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

Проход 1: Поиск минимума

8	↓	↓	8	5	6	7	4	10	3	1	2
			8	5	6	7	4	10	3	1	2
			8	5	6	7	4	10	3	1	2
			8	5	6	7	4	10	3	1	2
			8	5	6	7	4	10	3	1	2
			8	5	6	7	4	10	3	1	2
			8	5	6	7	4	10	3	1	2
			8	5	6	7	4	10	3	1	2

Обмен

1	5	6	7	4	10	3	8	2
---	---	---	---	---	----	---	---	---

↓ фиксированная позиция  
↓ переменная позиция  
(смещение вправо)

# Сортировка выбором

Проход 2: Поиск минимума

7 {

1	5	6	7	4	10	3	8	2
1	5	6	7	4	10	3	8	2
1	5	6	7	4	10	3	8	2
1	5	6	7	4	10	3	8	2
1	5	6	7	4	10	3	8	2
1	5	6	7	4	10	3	8	2
1	5	6	7	4	10	3	8	2

Обмен

1	2	6	7	4	10	3	8	5
---	---	---	---	---	----	---	---	---

Проход 3: Поиск минимума

6 {

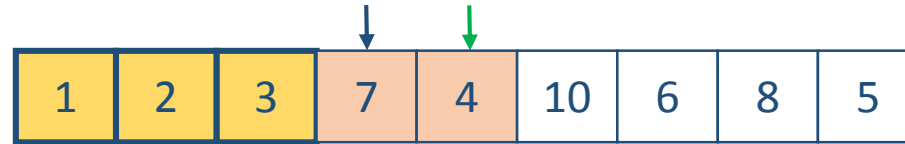
1	2	6	7	4	10	3	8	5
1	2	6	7	4	10	3	8	5
1	2	6	7	4	10	3	8	5
1	2	6	7	4	10	3	8	5
1	2	6	7	4	10	3	8	5
1	2	6	7	4	10	3	8	5

Обмен

1	2	3	7	4	10	6	8	5
---	---	---	---	---	----	---	---	---

# Сортировка выбором

**Проход 4:** Поиск минимума



Обмен



**Проход 8:**



# Сложность сортировки выбором

Сложность	Наилучший случай	Средний случай	Наихудший случай
Вычисление	$O(n)$	$O(n^2)$	$O(n^2)$
Память	$O(1)$	$O(1)$	$O(1)$

# Сортировка вставкой

# Сортировка вставкой

Входной список/массив:

8	5	6	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

**Проход 1:**

SortedList

SortedList								
		↓						
8	5	6	7	4	10	3	1	2
5	8	6	7	4	10	3	1	2

Сдвиг курсора

Вставка/перемещение

SortedList – отсортированная часть списка/массива

**Проход 2:**

SortedList

SortedList								
			↓					
5	8	6	7	4	10	3	1	2
5	6	8	7	4	10	3	1	2

Сдвиг курсора

Вставка/перемещение

**Проход 3:**

SortedList

SortedList								
				↓				
5	6	8	7	4	10	3	1	2
5	6	7	8	4	10	3	1	2

Сдвиг курсора

Вставка/перемещение

**Проход 4:**

SortedList

SortedList								
					↓			
5	6	7	8	4	10	3	1	2
4	5	6	7	8	10	3	1	2

Сдвиг курсора

Вставка/перемещение

**Проход 8:**

				↓				
1	2	3	4	5	6	7	8	10



# Сортировка вставкой

Входной список/массив:

8	5	6	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

**Проход 1:** Смещение на 1 вправо

8	5	6	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

↓ фиксированная позиция

Перемещение

8	5	6	7	4	10	3	1	2
8	8	6	7	4	10	3	1	2
5	8	6	7	4	10	3	1	2

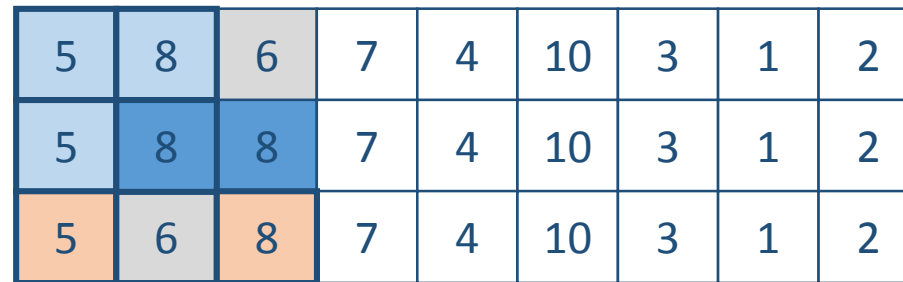
# Сортировка вставкой

**Проход 2:** Смещение на 1 вправо



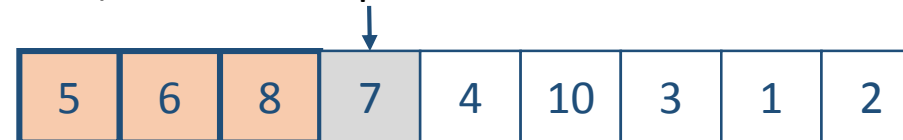
5	8	6	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

Перемещение



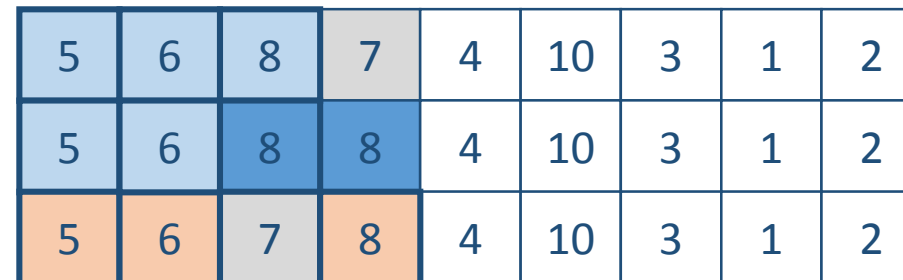
5	8	6	7	4	10	3	1	2
5	8	8	7	4	10	3	1	2
5	6	8	7	4	10	3	1	2

**Проход 3:** Смещение на 1 вправо



5	6	8	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---

Перемещение



5	6	8	7	4	10	3	1	2
5	6	8	8	4	10	3	1	2
5	6	7	8	4	10	3	1	2

# Сортировка вставкой

**Проход 4:** Смещение на 1 вправо

5	6	7	8	4	10	3	1	2
---	---	---	---	---	----	---	---	---

Перемещение

5	6	7	8	4	10	3	1	2
5	6	7	8	8	10	3	1	2
5	6	7	7	8	10	3	1	2
5	6	6	7	8	10	3	1	2
5	5	6	7	8	10	3	1	2
4	5	6	7	8	10	3	1	2

**Проход 5:** Смещение на 1 вправо

4	5	6	7	8	10	3	1	2
---	---	---	---	---	----	---	---	---

Перемещение

4	5	6	7	8	10	3	1	2
---	---	---	---	---	----	---	---	---

# Сортировка вставкой

**Проход 6:** Поиск минимума

4	5	6	7	8	10	3	1	2
---	---	---	---	---	----	---	---	---

Перемещение

4	5	6	7	8	10	3	1	2
4	5	6	7	8	10	10	1	2
4	5	6	7	8	8	10	1	2
4	5	6	7	7	8	10	1	2
4	5	6	6	7	8	10	1	2
4	5	5	6	7	8	10	1	2
4	4	5	6	7	8	10	1	2
3	4	5	6	7	8	10	1	2

**Проход 8:**

1	2	3	4	5	6	7	8	10
---	---	---	---	---	---	---	---	----

# Сложность сортировки вставкой

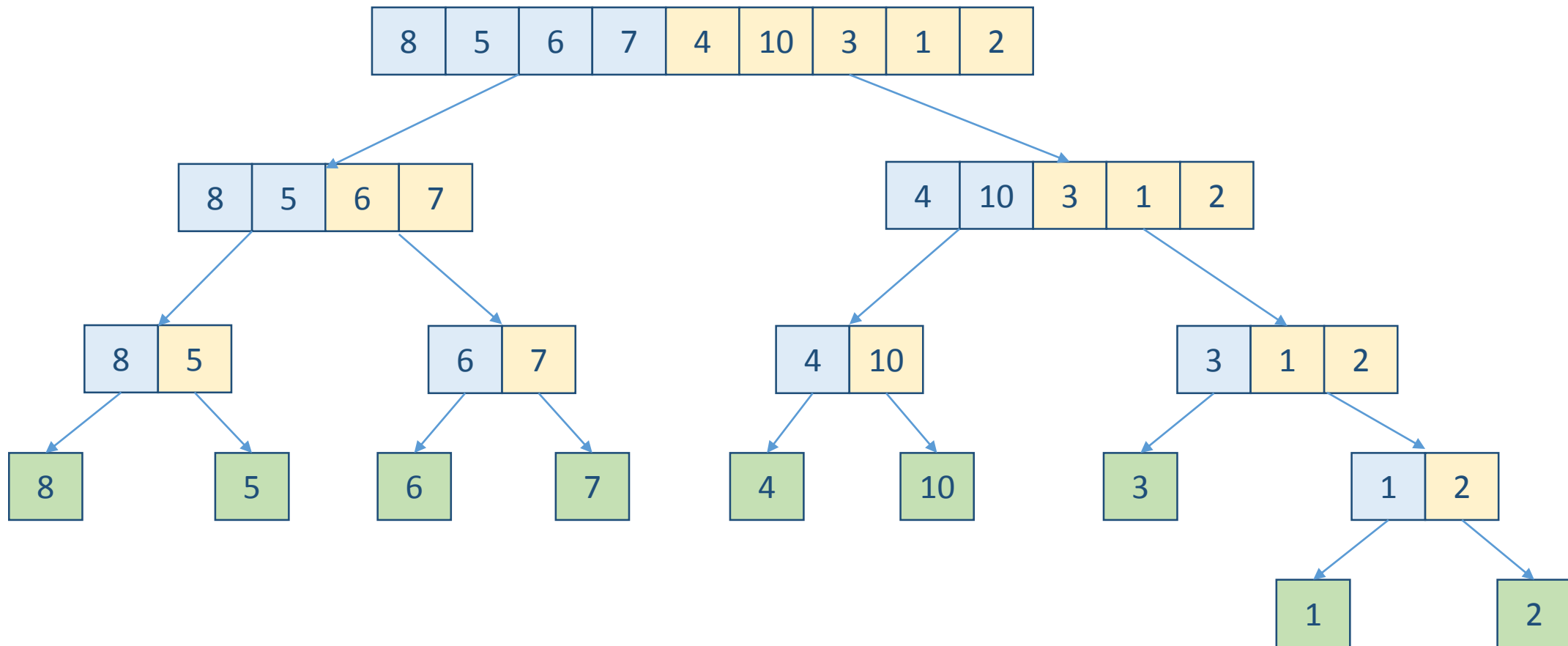
Сложность	Наилучший случай	Средний случай	Наихудший случай
Вычисление	$O(n)$	$O(n^2)$	$O(n^2)$
Память	$O(1)$	$O(1)$	$O(1)$

# Сортировка слиянием

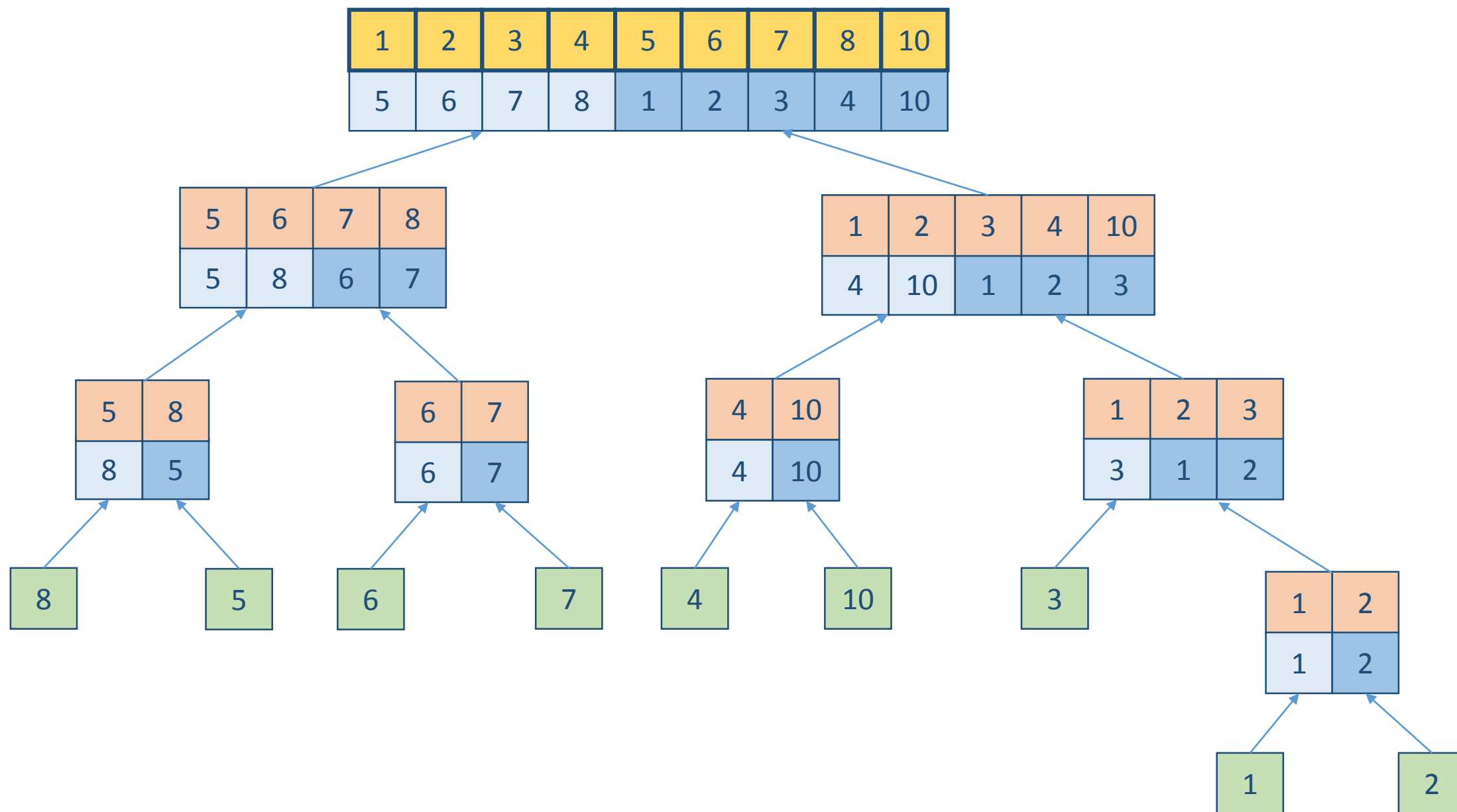
# Сортировка слиянием

Входной список/массив:

8	5	6	7	4	10	3	1	2
---	---	---	---	---	----	---	---	---



# Сортировка слиянием





# Сортировка слиянием

5

## Шаг 1

### Шаг 3

5

8



10

# 2

## Шаг 1

### Шаг 3

5

8

E

—

1041

100

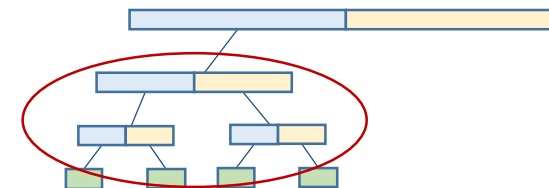
1041



2

# Сортировка слиянием

5	8	6	7
---	---	---	---

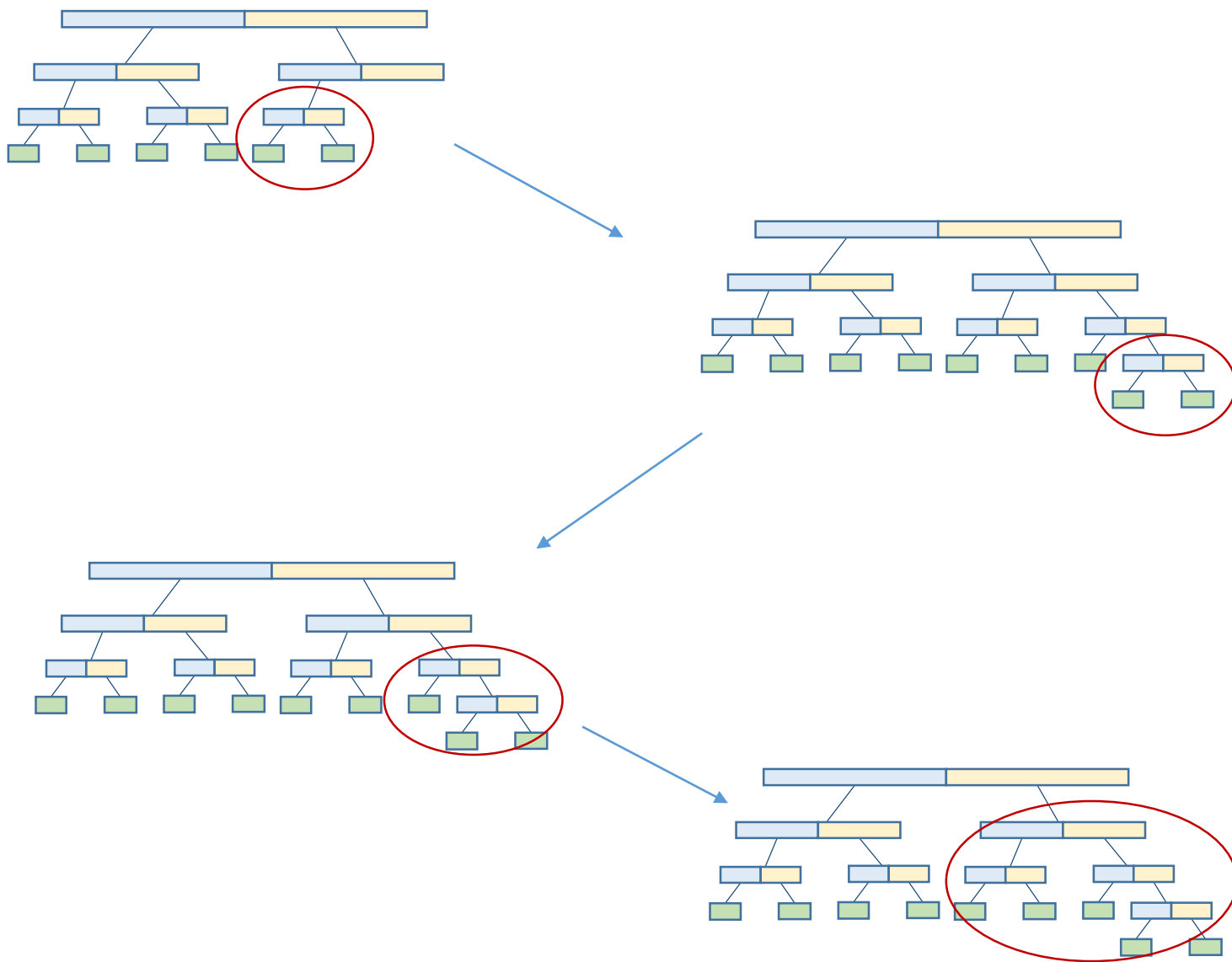


Шаг 4	5	8	6	7	4	10	3	1	2
Шаг 3	5	8	6	7	4	10	3	1	2
Шаг 2	5	8	6	7	4	10	3	1	2
Шаг 1	5	8	6	7	4	10	3	1	2

Шаг 5

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

# Сортировка слиянием

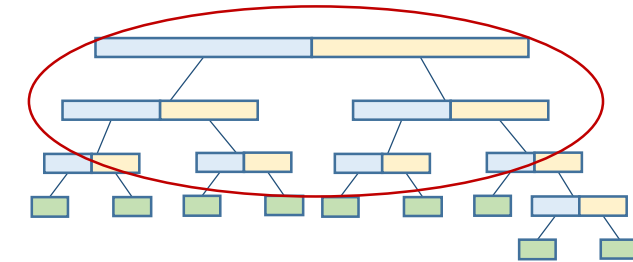


# Сортировка слиянием

5	6	7	8
---	---	---	---

1	2	3	4	10
---	---	---	---	----

1	2	3	4	5	6	7	8	10
---	---	---	---	---	---	---	---	----



Шаг 9	5	6	7	8	1	2	3	4	10	→	1	2	3	4	5	6	7	8	10
Шаг 8	5	6	7	8	1	2	3	4	10	→	1	2	3	4	5	6	7	8	0
Шаг 7	5	6	7	8	1	2	3	4	10	→	1	2	3	4	5	6	7	0	0
Шаг 6	5	6	7	8	1	2	3	4	10	→	1	2	3	4	5	6	0	0	0
Шаг 5	5	6	7	8	1	2	3	4	10	→	1	2	3	4	5	0	0	0	0
Шаг 4	5	6	7	8	1	2	3	4	10	→	1	2	3	4	0	0	0	0	0
Шаг 3	5	6	7	8	1	2	3	4	10	→	1	2	3	0	0	0	0	0	0
Шаг 2	5	6	7	8	1	2	3	4	10	→	1	2	0	0	0	0	0	0	0
Шаг 1	5	6	7	8	1	2	3	4	10	→	1	0	0	0	0	0	0	0	0
											0	0	0	0	0	0	0	0	0

# Сложность сортировки слиянием

Сложность	Наилучший случай	Средний случай	Наихудший случай
Вычисление	$O(n \cdot \log n)$	$O(n \cdot \log n)$	$O(n \cdot \log n)$
Память	$O(n)$	$O(n)$	$O(n)$

# Сортировка timsort