



# Сортировка пузырьком



- 1) Сортировка пузырьком
- 2) Сортировка пузырьком с условием Айверсона – 1
- 3) Сортировка пузырьком с условием Айверсона – 2



# СОРТИРОВКА МЕТОДОМ ПУЗЫРЬКА

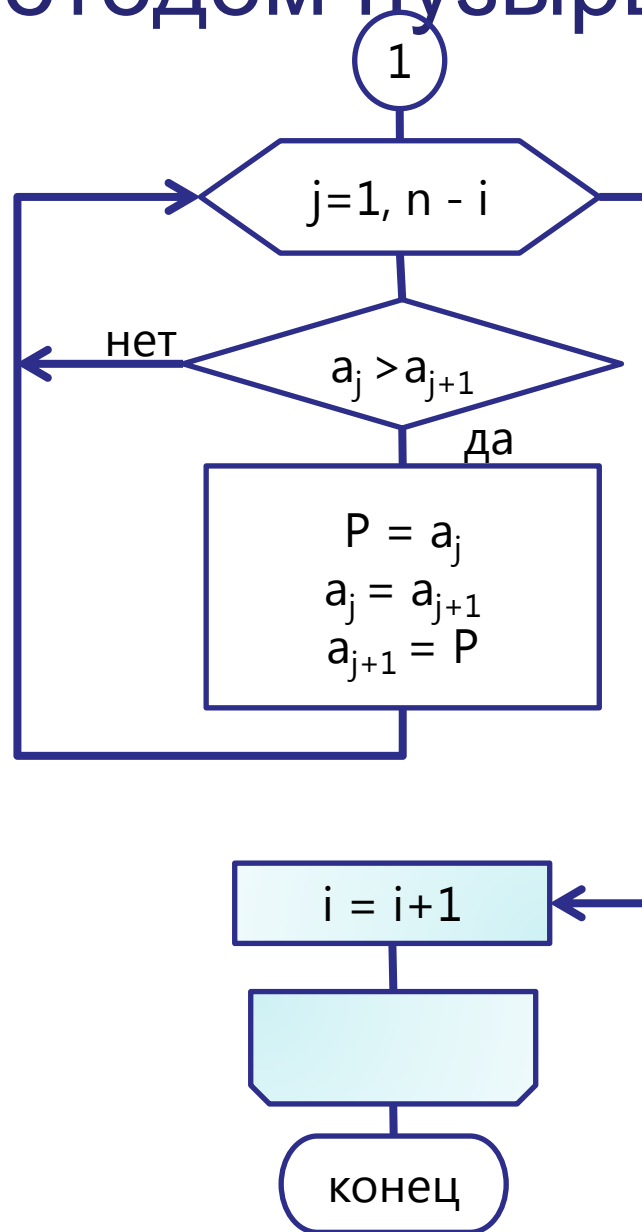
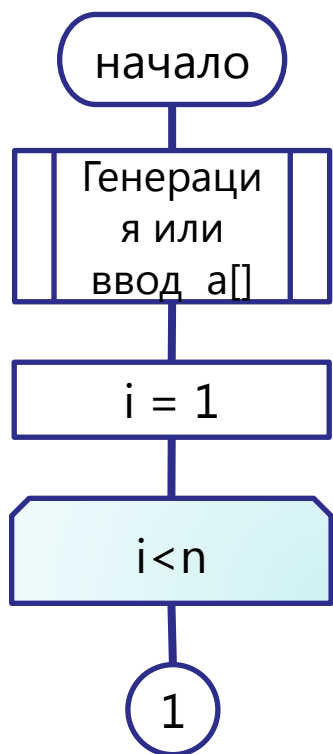
Будем просматривать последовательность  $N-1$  раз и сравнивать **текущий элемент со следующим**, меняя их местами, если следующий больше.

После каждого просмотра максимальный элемент занимает свое место в упорядоченной последовательности

В ходе следующего просмотра обрабатывается на один элемент меньше.



# Сортировка методом пузырька





# Пример

	<i>j</i>		1	2	3	4	5	6
1	1		4	5	9	1	3	6
	2		4	5	9	1	3	6
	3	обмен	4	5	9	1	3	6
	4	обмен	4	5	1	9	3	6
	5	обмен	4	5	1	3	9	6
2	1		4	5	1	3	6	9
	2	обмен	4	5	1	3	6	9
	3	обмен	4	1	5	3	6	9
	4		4	1	3	5	6	9
3	1	обмен	4	1	3	5	6	9
	2	обмен	1	4	3	5	6	9
	3		1	3	4	5	6	9
4	1		1	3	4	5	6	9
	2		1	3	4	5	6	9
5	1		1	3	4	5	6	9



# Замечание

Если вы написали при решении задачи 3 (день 1) программу сортировки НЕ пузырьком, срочно выполните следующее

Написать программу сортировки по возрастанию целочисленного массива методом пузырька.

На **вход** подаем (с клавиатуры или в коде (случайные числа))

- количество элементов массива N
- элементы массива

Сами понимаете, массив динамический

**Вывод** на консоль

- Сортированный массив

**Отчетность** – проверка ассистентами на семинаре

И загрузите в ЛМС задание 3 еще раз в проект Cpp3



**Кеннет** Юджин **Айверсон** (англ. **Kenneth Eugene Iverson**; 17 декабря 1920 — 19 октября 2004, Канада)

— канадский учёный в области теории вычислительных систем, программист, автор языка программирования APL

# ПУЗЫРЕК С УСЛОВИЕМ АЙВЕРСОНА \_ 1



# Пример

<i>i</i>	<i>j</i>		1	2	3	4	5	6
1	1		4	5	9	1	3	6
	2		4	5	9	1	3	6
	3	обмен	4	5	9	1	3	6
	4	обмен	4	5	1	9	3	6
	5	обмен	4	5	1	3	9	6
2	1		4	5	1	3	6	9
	2	обмен	4	5	1	3	6	9
	3	обмен	4	1	5	3	6	9
	4		4	1	3	5	6	9
3	1	обмен	4	1	3	5	6	9
	2	обмен	1	4	3	5	6	9
	3		1	3	4	5	6	9
4	1		1	3	4	5	6	9
	2		1	3	4	5	6	9
5	1		1	3	4	5	6	9

Нет обменов

Не выполняем

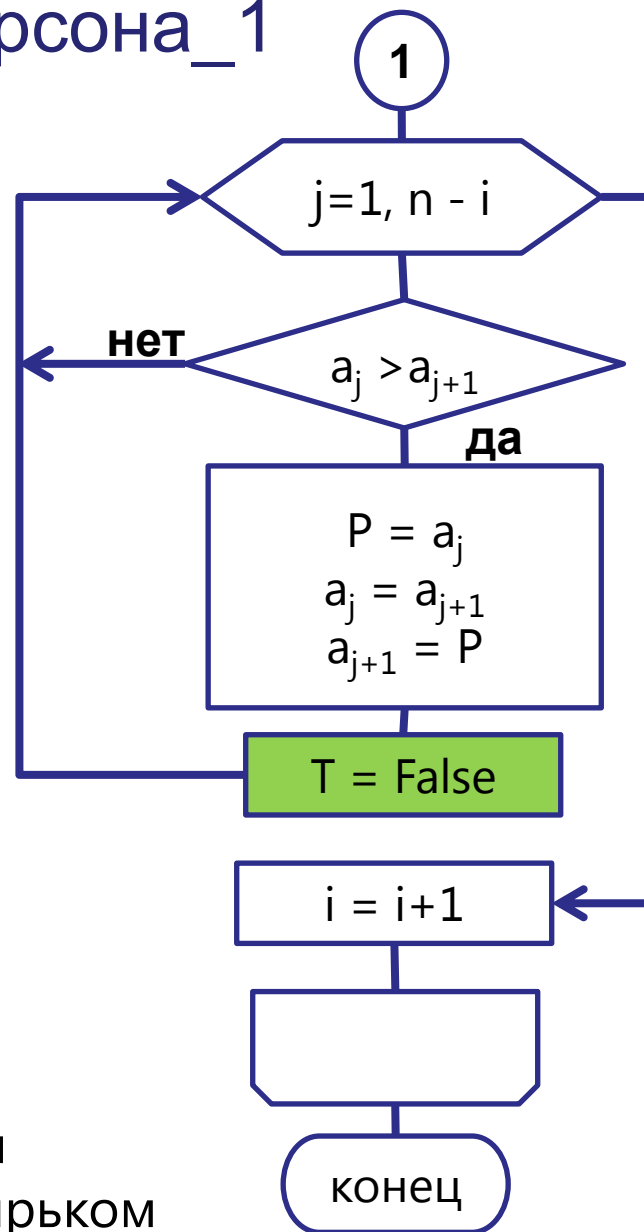
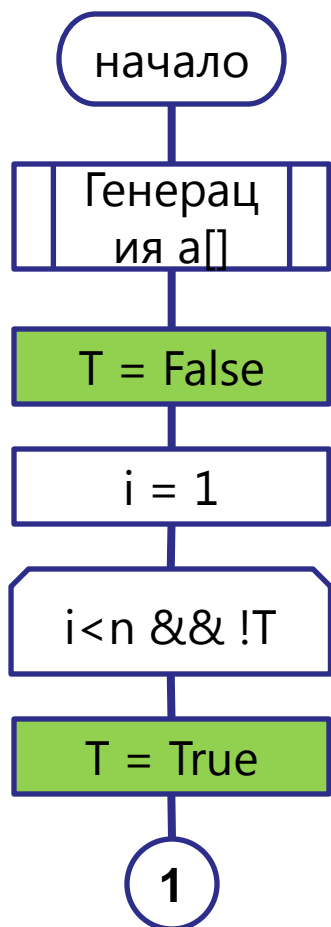
Массив уже отсортирован, но алгоритм пузырька все равно будет работать.

Проверка условия Айверсона позволяет не продолжать сортировку, если массив уже отсортирован. Если не было ни одного обмена,

**выходим из внешнего цикла**



# Сортировка методом пузырька с условием Айверсона\_1



Зеленым цветом выделены фрагменты алгоритма – добавки к алгоритму пузырьком





# Основной результат

Сократили количество проходов внешнего цикла



**Кеннет** Юджин **Айверсон** (англ. **Kenneth Eugene Iverson**; 17 декабря 1920 — 19 октября 2004, Канада) — канадский учёный в области теории вычислительных систем, программист, автор языка программирования APL

# ПУЗЫРЕК С УСЛОВИЕМ АЙВЕРСОНА \_ 2



# Сортировка методом пузырька с условием Айверсона\_2

Запоминаем, в какой позиции  $t$  был последний обмен на предыдущей итерации внешнего цикла.

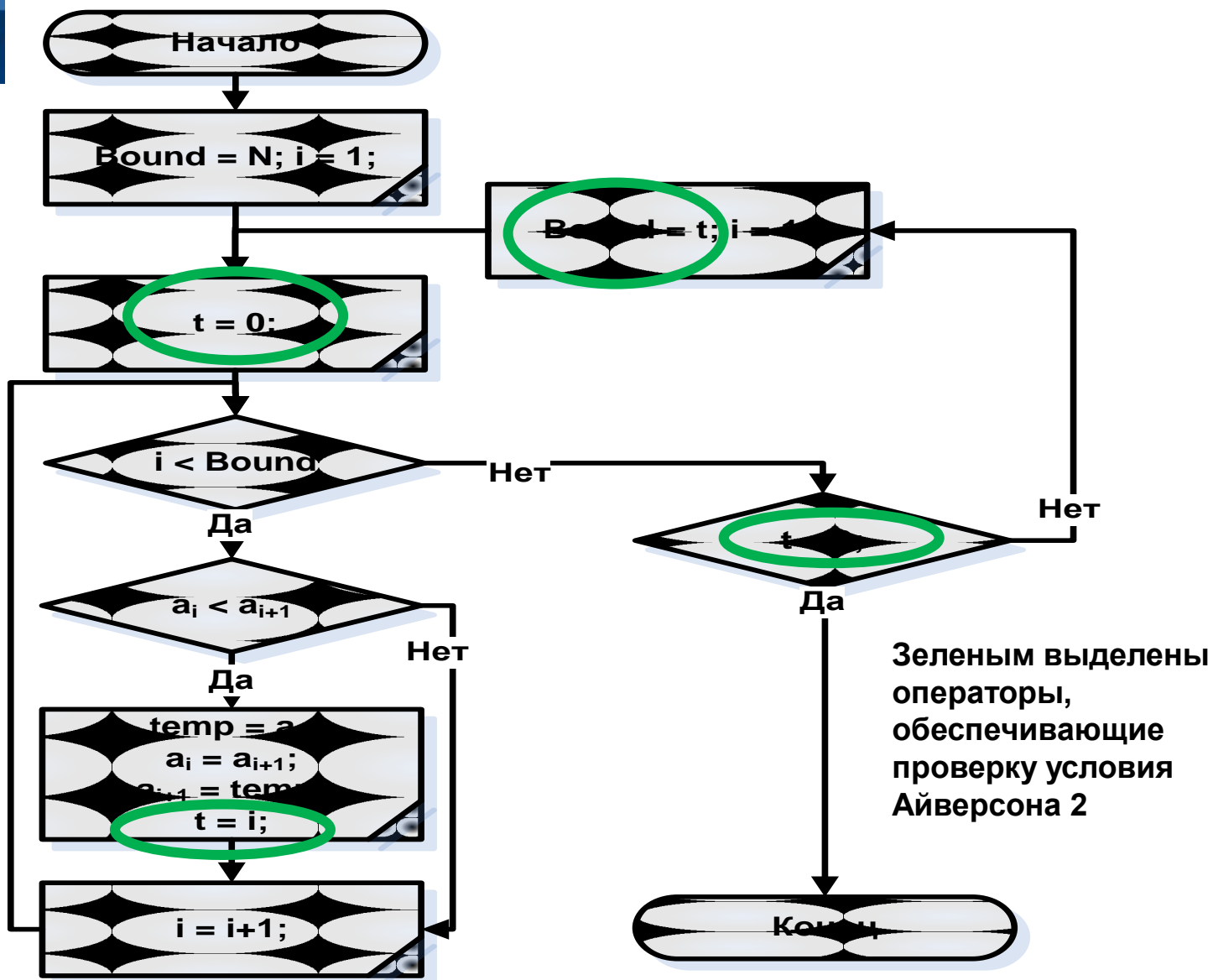
Это – верхняя граница просмотра массива Bound на след. итерации

Если  $t=0$  после выполнения внутреннего цикла, значит, обменов не было, алгоритм заканчивает работу.

Основная идея – **уменьшаем количество проходов внутреннего цикла**



# Сортировка методом пузырька с условием Айверсона\_2





# Генератор случайных чисел

```
#include <ctime>
```

```
...
```

```
srand(time(0));
```

```
...
```

```
for (int i = 0; i < num; i++)
```

```
a[i] = min1+rand()%(max1 - min1 + 1);
```

позволяет при каждом запуске программы сгенерировать новую последовательность псевдослучайных чисел.

Например, сгенерировать случайные числа из диапазона [-10;20]

$-10 + \text{rand()} \%(20 + 10 + 1);$

$\text{rand()} \%(20 + 10 + 1)$

