

# Генерация случайного числа

`void rand ();`

- Возвращает случайное целое число в диапазоне от 0 до RAND\_MAX (0x7FFF), т.е. от 0 до 32767.
- Число генерируется алгоритмом, который при каждом вызове возвращает последовательность явно не связанных чисел.
- Алгоритм использует начальное число, заданное с помощью функции **srand**.

# Заполнение массива случайными числами

```
srand(time(0));  
int arr[10];  
for (int i = 0; i < 10; i++)  
{  
    arr[i] = rand() % 31;        //в интервале[0; 30]  
    //arr[i] = rand() % 31 + 5 ;    //в интервале[5; 35]  
}
```

//Формула для генерации случайных чисел из заданного  
//отрезка [A, B] :  $R = A + \text{rand}() \% (B - A + 1);$

# Заполнение массива случайными числами

Иногда требуется заполнить массив неповторяющимися числами из заданного отрезка натурального ряда, расположенными в случайном порядке.

```
const int N=100;
```

```
int A[N];
```

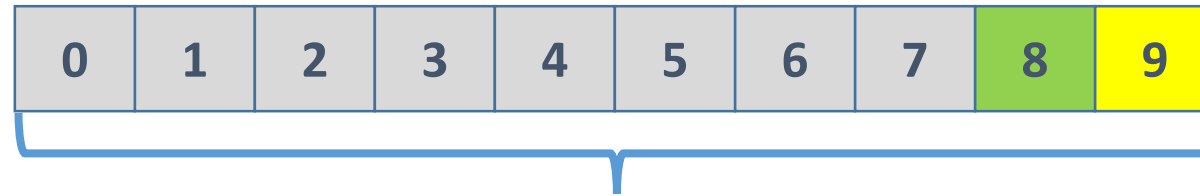
```
//Например, массив надо заполнить числами из [0, 99]
```

```
//это частный случай
```

# Алгоритм «Тасование колоды»

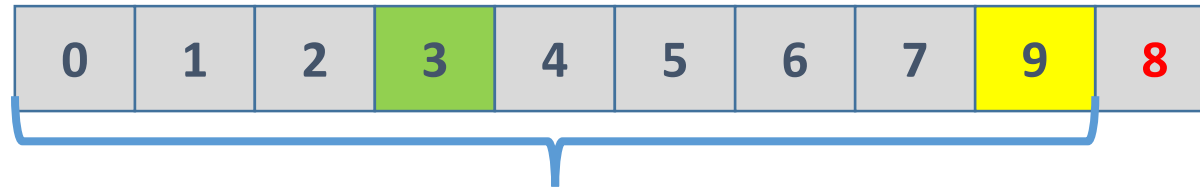
- Массив  $A[N]$  инициализируется последовательными значениями из отрезка, например  $[0, N-1]$
- $index = N-1$ , где  $index$ - максимальное значение индекса.
- В цикле по  $index$  выполняется перемешивание значений:
  1. Генерируется случайное число  $R$  из отрезка  $[0, index]$
  2. Выполняется обмен значений  $A[index]$  и  $A[R]$
  3.  $index = index - 1$
  4. Если  $index == 0$ , то конец
  5. Переход к шагу 1

8



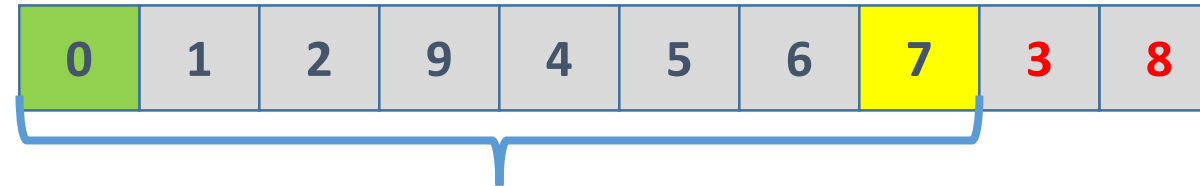
index=N-1  
R=rand() % (index+1)  
ar[index] <=> ar [R]

3



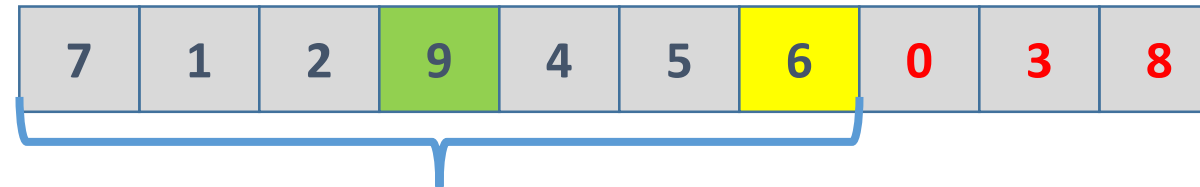
index=index-1  
R=rand() % (index+1)  
ar[index] <=> ar [R]

0

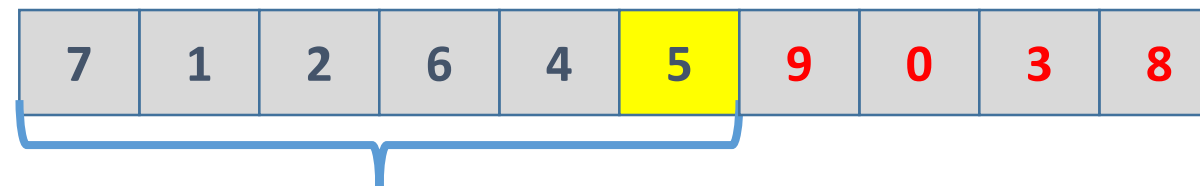


index=index-1  
R=rand() % (index+1)  
ar[index] <=> ar [R]

3



index=index-1  
R=rand() % (index+1)  
ar[index] <=> ar [R]



index=index-1  
R=rand() % (index+1)  
ar[index] <=> ar [R]

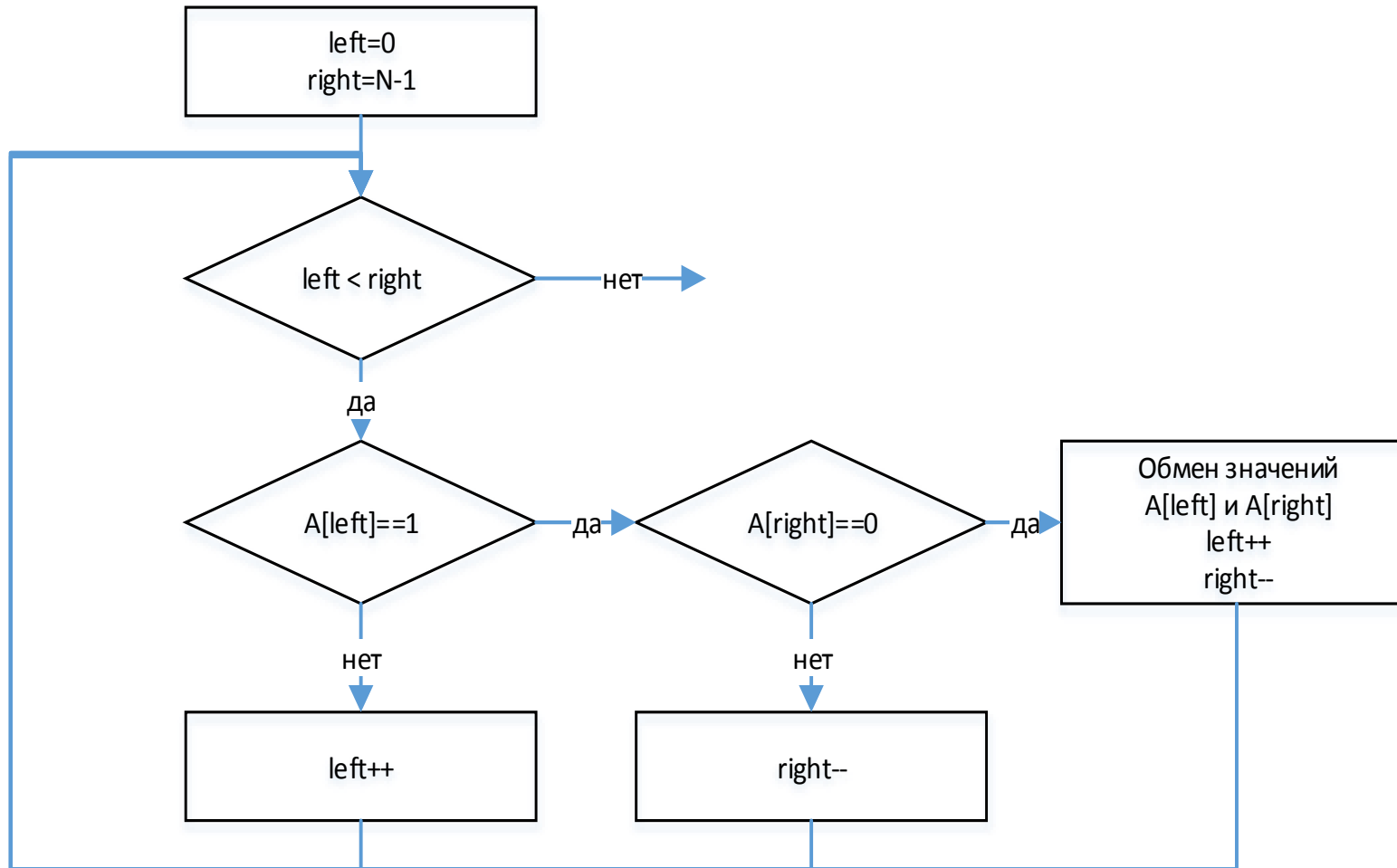
# Задача (псевдосортировка)

Массив из  $N$  элементов случайным образом заполнен нулями и единицами. Переупорядочить элементы массива так, чтобы сначала шли нули, а потом единицы.

## **Очевидное решение (два цикла):**

- использовать последовательный перебор
- подсчитать число нулей (или единиц) в массиве.
- При повторном проходе по массиву в соответствующее число ячеек записываются нули, а в остальные – единицы.

# Задача (псевдосортировка). 1 способ



1

**1** 1 0 0 0 1 0 1 0 0 0 1 1 0 **1**

**1** 1 0 0 0 1 0 1 0 0 0 1 1 **0** 1

обмен

**0** 1 0 0 0 1 0 1 0 0 0 1 1 **1** 1

0 **1** 0 0 0 1 0 1 0 0 **0** 1 1 1 1

обмен

0 **0** 0 0 0 1 0 1 0 0 **1** 1 1 1 1

0 0 0 0 0 **1** 0 1 0 **0** 1 1 1 1 1

обмен

0 0 0 0 0 **0** 0 1 0 **1** 1 1 1 1 1

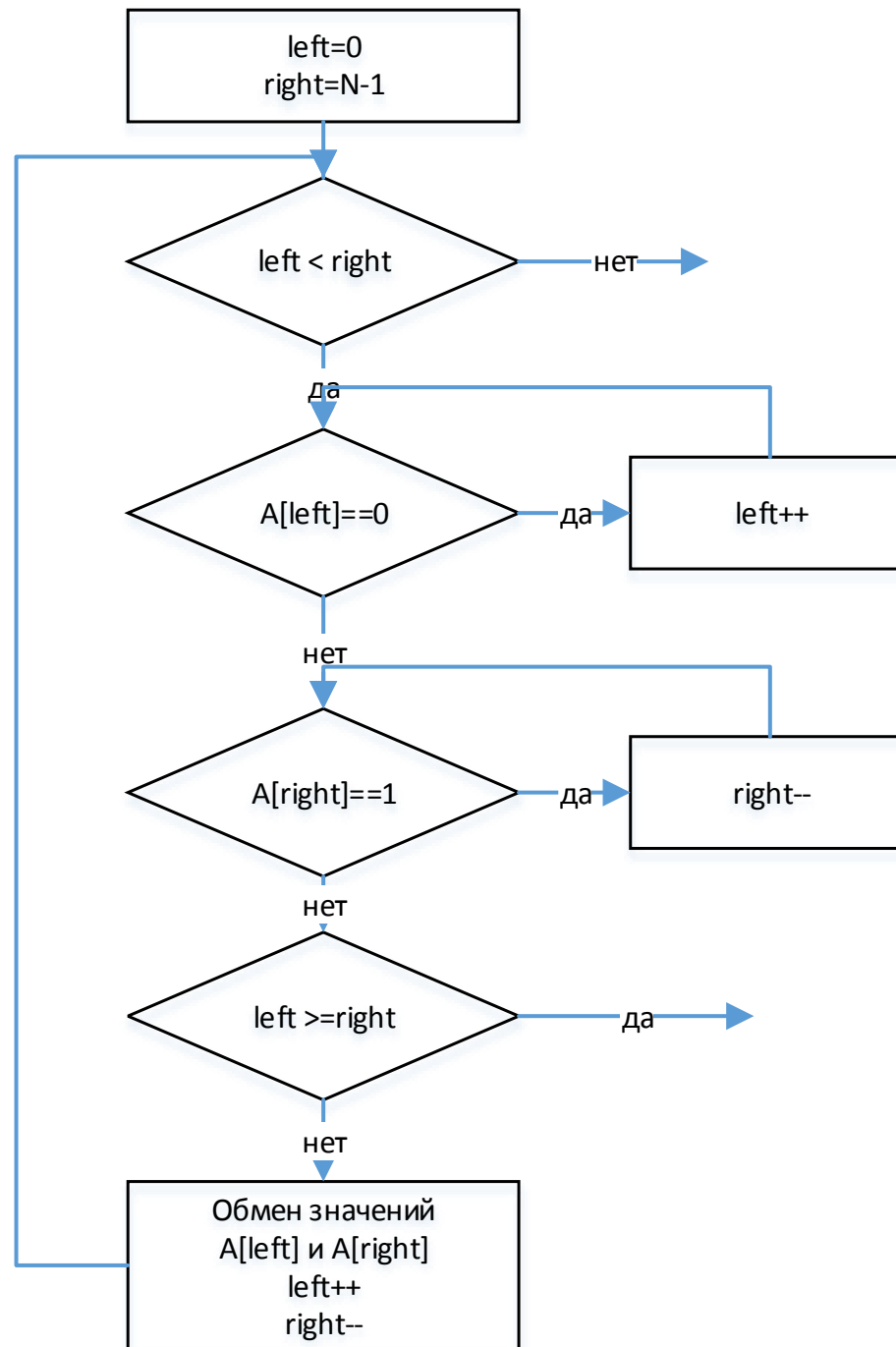
0 0 0 0 0 0 0 **1** **0** 1 1 1 1 1 1

Обмен

0 0 0 0 0 0 0 0 1 1 1 1 1 1 1



## 2 способ



2

0	0	0	1	0	1	0	1	0	0	0	1	1	0	1
0	0	0	1	0	1	0	1	0	0	0	1	1	0	1
обмен														
0	0	0	0	0	1	0	1	0	0	0	1	1	1	1
0	0	0	0	0	1	0	1	0	0	0	1	1	1	1
Обмен														
0	0	0	0	0	0	0	1	0	0	1	1	1	1	1
0	0	0	0	0	0	0	1	0	0	1	1	1	1	1
Обмен														
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
КОНЕЦ РАБОТЫ														
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1