

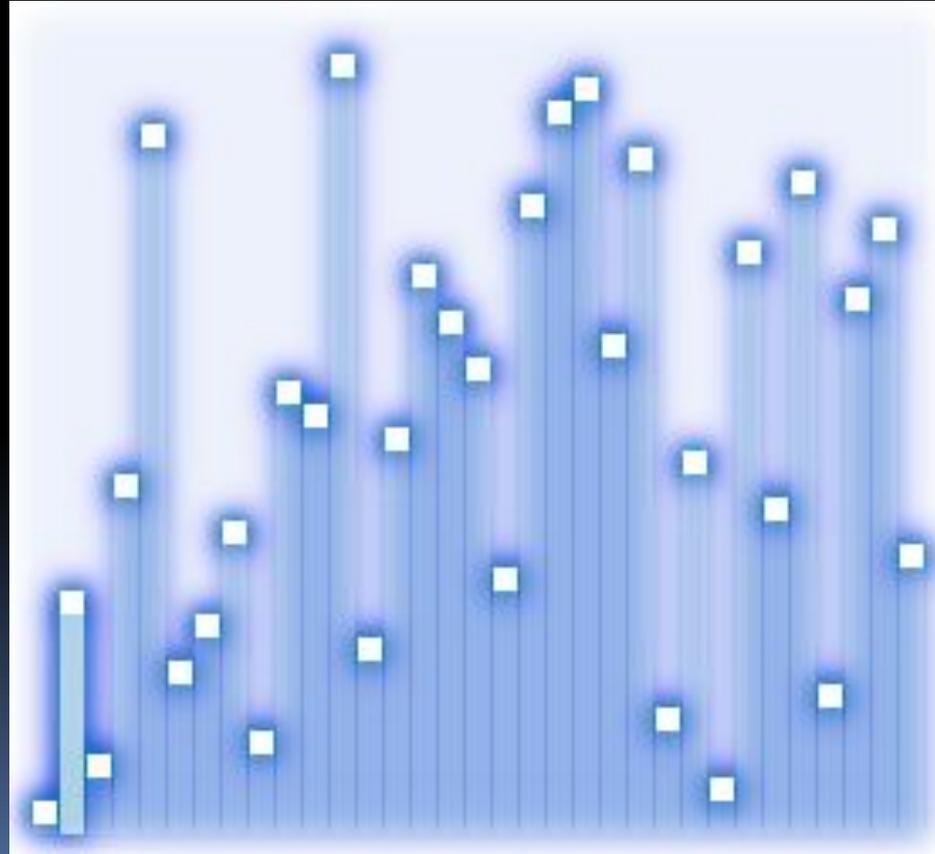


# ШЕЙКЕРНАЯ СОРТИРОВКА (ПЕРЕМЕШИВАНИЕМ)

Выполнил(а): студент гр. ПИ620-И2  
Бернгардт Геннадий Сергеевич  
Проверил: доцент кафедры ПИЭ  
Пестова Светлана Юрьевна

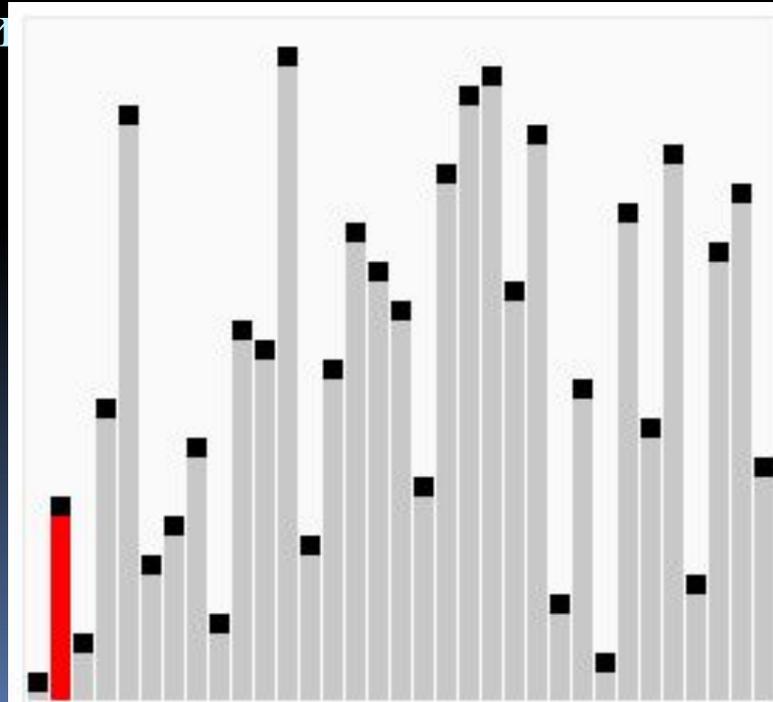
Омск 2020

**Шейкерная сортировка (cocktail sort, shaker sort),**  
или **сортировка перемешиванием** - усовершенствованная  
разновидность **сортировки пузырьком**, при  
которой **сортировка** производится в двух направлениях,  
меняя направление при каждом проходе.



- 
- Рассматриваемый алгоритм имеет несколько непохожих друг на друга названий. Среди них: сортировка перемешиванием, двунаправленная пузырьковая сортировка, шейкерная сортировка, пульсирующая сортировка (ripple sort), трансфертная сортировка (shuttle sort), и даже сортировка «счастливого часа» (happy hour sort).

Второй вариант (двунаправленная пузырьковая сортировка) наиболее точно описывает процесс работы алгоритма. Здесь, в его название, довольно-таки удачно включен термин «пузырьковая». Это действительно альтернативная версия известного метода, модификации в котором заключаются, по большей части, в реализации, упомянутой в названии, двунаправленности: алгоритм перемещается, ни как в обменной (пузырьковой) сортировке – строго снизу вверх (слева направо), а сначала снизу вверх, потом сверху вниз.



5	8	7	1	14	2	13	9	11	6	10	15	12	3	4
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

00:54

- ❖ Перестановка элементов в шейкерной сортировке выполняется аналогично той же в пузырьковой сортировке, т. е. два соседних элемента, при необходимости, меняются местами. Пусть массив требуется упорядочить по возрастанию. Обозначим каждый пройденный путь от начала до конца последовательности через  $W_i$ , где  $i$  – номер пути; а обратный путь (от конца к началу) через  $\leftarrow W_j$ , где  $j$  – номер пути.
- ❖ Тогда после выполнения  $W_i$ , один из неустановленных элементов будет помещен в позицию справа, как наибольший из еще неотсортированных элементов, а после выполнения  $\leftarrow W_j$ , наименьший из неотсортированных, переместится в некоторую позицию слева. Так, например, после выполнения  $W_1$  в конце массива окажется элемент, имеющий наибольшее значение, а после  $\leftarrow W_1$  в начало отправится элемент с наименьшим значением.

# Примеры записи кода:

## Код программы на C++

```
1 #include "stdafx.h"
2 #include <iostream>
3 using namespace std;
4 //функция обмена
5 void Swap(int *Mas, int i)
6 {
7     int temp;
8     temp=Mas[i];
9     Mas[i]=Mas[i-1];
10    Mas[i-1]=temp;
11 }
12 //функция шейкерной сортировки
13 void ShakerSort(int *Mas, int Start, int N)
14 {
15     int Left, Right, i;
16     Left=Start;
17     Right=N-1;
18     while (Left<=Right)
19     {
20         for (i=Right; i>=Left; i--)
21             if (Mas[i-1]>Mas[i]) Swap(Mas, i);
22         Left++;
23         for (i=Left; i<=Right; i++)
24             if (Mas[i-1]>Mas[i]) Swap(Mas, i);
25         Right--;
26     }
27 }
28 //главная функция
29 void main()
30 {
31     setlocale(LC_ALL, "Rus");
32     int n, k;
33     cout<<"Размер массива > "; cin>>n;
34     int *A=new int[n];
35     for (k=0; k<n; k++)
36         { cout<<k+1<<" элемент > "; cin>>A[k]; }
37     ShakerSort(A, 1, n);
38     cout<<"Результирующий массив: ";
39     for (k=0; k<n; k++)cout<<" "<<A[k];
40     system("pause>>void");
41 }
```

# Примеры записи кода:

## Код программы на Pascal

```
1 program CocktailSort;
2 uses crt;
3 type Massiv=array[1..100] of integer;
4 var
5 n, k: integer;
6 A: Massiv;
7 {процедура шейкерной сортировки}
8 procedure ShakerSort(Mas: Massiv; Start, m: integer);
9 var Left, Right, temp, i: integer;
10 begin
11 Left:=Start;
12 Right:=m;
13 while Left<=Right do
14 begin
15 for i:=Right downto Left do
16 if (Mas[i-1]>Mas[i]) then
17 begin
18 temp:=Mas[i];
19 Mas[i]:=Mas[i-1];
20 Mas[i-1]:=temp;
21 end;
22 Left:=Left+1;
23 for i:=Left to Right do
24 if Mas[i-1]>Mas[i] then
25 begin
26 temp:=Mas[i];
27 Mas[i]:=Mas[i-1];
28 Mas[i-1]:=temp;
29 end;
30 Right:=Right-1;
31 end;
32 for i:=1 to M do write(' ', Mas[i]);
33 end;
34 {основной блок программы}
35 begin
36 clrscr;
37 write('Размер массива > '); read(n);
38 for k:=1 to n do
39 begin
40 write(k, ' элемент > '); read(A[k]);
41 end;
42 write('Результирующий массив: ');
43 ShakerSort(A, 2, n);
44 end.
```



*Спасибо за внимание!*