



# Алгоритмы и структуры данных

## Лекция 5. Простые методы сортировки

---

Антон Штанюк (к.т.н, доцент)

10 марта 2021 г.

Нижегородский государственный технический университет им. Р.Е. Алексеева  
Институт радиоэлектроники информационных технологий  
Кафедра "Компьютерные технологии в проектировании и производстве"

Виды сортировок

Непрактичная сортировка

Сортировка вставками

Сортировка выбором

Пузырьковая сортировка

Итоги

Список литературы

## Виды сортировок

---

## Определение

**Сортировка** - процесс упорядочивания данных по возрастанию (убыванию) их значений.

Простейшие методы имеют оценку сложности  $O(n^2)$ . Если  $n$  невелико, то такие алгоритмы вполне могут считаться эффективными, благодаря простоте реализации. Но с ростом числа элементов будет неизбежно возрастать время сортировки, поэтому при больших  $n$  предпочтение нужно отдать более "продвинутым" алгоритмам.

К простейшим алгоритмам сортировки можно отнести:

- непрактичную сортировку;
- сортировку вставками;
- сортировку выбором;
- пузырьковую сортировку.

К более продвинутым можно отнести:

- быструю сортировку;
- сортировку Шелла;
- пирамидальную сортировку;
- сортировку слиянием;
- сортировку с помощью бинарного дерева.

Практически во всех методах сортировки есть необходимость обмена местами элементов. Этот обмен можно реализовать разными способами, но для упрощения кода мы вынесем его в отдельную функцию **swap**.

```
void swap(int *a,int *b)
{
    int t=*a;
    *a=*b;
    *b=t;
}
```

## Непрактичная сортировка

---

Самой простой для запоминания является непрактичная сортировка.

```
void UnratioSort(int arr[], int N)
{
    int i, j;
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            if(arr[i]<arr[j])
                swap(&arr[i], &arr[j]);
}
```

Сортировка может считаться непрактичной из-за лишних операций сравнения и обмена (например, arr[0] с arr[0]).

## Сортировка вставками

---

Сортировка вставками удобна для сортировки коротких последовательностей элементов. Именно таким образом обычно сортируют карты: держа в левой руке уже упорядоченные карты и взяв правой рукой очередную карту мы вставляем ее в нужное место, сравнивая с уже имеющимися, двигаясь справа налево.

Все элементы условно разделяются на готовую последовательность  $arr_1 \dots arr_{i-1}$  и входную  $arr_i \dots arr_N$ . На каждом шаге, начиная с  $i = 2$  и увеличивая  $i$  на 1, берем  $i$ -й элемент входной последовательности и вставляем его на нужное место в готовую.

```
44 \\ 55 12 42 94 18 06 67
44 55 \\ 12 42 94 18 06 67
12 44 55 \\ 42 94 18 06 67
12 42 44 55 \\ 94 18 06 67
12 42 44 55 94 \\ 18 06 67
12 18 42 44 55 94 \\ 06 67
06 12 18 42 44 55 94 \\ 67
06 12 18 42 44 55 67 94 \\
```

```
void InsertionSort(int arr[], int N)
{
    int i, j;
    int temp;
    for(i=1; i<N; i++)
    {
        j=i;
        temp=arr[i];
        while(j>0 && temp<arr[j-1])
        {
            arr[j]=arr[j-1];
            j--;
        }
        arr[j]=temp;
    }
}
```

## Сортировка выбором

---

Предполагается, что  $N$  элементов данных хранятся в массиве. и по этому массиву выполняется  $N - 1$  проход. В нулевом проходе выбирается наименьший элемент, который затем меняется местами с  $arr[0]$ . После этого неупорядоченными остаются элементы  $arr[1] \dots arr[N - 1]$ . В следующем проходе просматривается эта часть массива и наименьший элемент помещается в  $arr[1]$ . И так до конца.

```
void SelectionSort(int arr[], int N)
{
    int i,j,index;
    for(i=0;i<N-1;i++)
    {
        index=i;
        for(j=i+1;j<N;j++)
            if(arr[j]<arr[index])
                index=j;
        swap(&arr[i],&arr[index]);
    }
}
```

## Пузырьковая сортировка

---

Для сортировки  $N$ -элементного массива методом пузырька требуется до  $N - 1$  проходов. В каждом проходе сравниваются соседние элементы, и если, первый из них больше или равен второму, эти элементы меняются местами.

Можно выделить два варианта реализации алгоритма:

1. избыточный;
2. экономный.

```
void BubbleSort(int arr[], int N)
{
    int i, j;
    for(i=0; i<N-1; i++)
        for(j=N-1; j>i; j--)
            if(arr[j-1]>arr[j])
                swap(&arr[j-1], &arr[j]);
}
```

В избыточном варианте циклы выполняются независимо от начального расположения элементов, что ведёт к лишним операциям сравнения, если массив уже упорядочен.

```
void BubbleSort2(int arr[], int N)
{
    int i,j,lastIndex;
    i=N-1;
    while(i>0)
    {
        lastIndex=0;
        for(j=0;j<i;j++)
            if(arr[j+1]<arr[j])
            {
                swap(&arr[j+1],&arr[j]);
                lastIndex=j;
            }
        i=lastIndex;
    }
}
```

## Итоги

---

К положительным качествам простейших методов сортировки относят

- экономное расходование памяти
- простота реализации

Главным недостатком сортировок методом вставок, выбора, пузырька является низкая производительность

Количество операций		
<i>Метод сортировки</i>	<i>Тип операции</i>	<i>Количество</i>
Selection	сравнение обмен	$N^2/2$ $N$
Insertion	сравнение обмен	$N^2/4$ $N^2/4$
Bubble	сравнение обмен	$N^2/2$ $N^2/2$

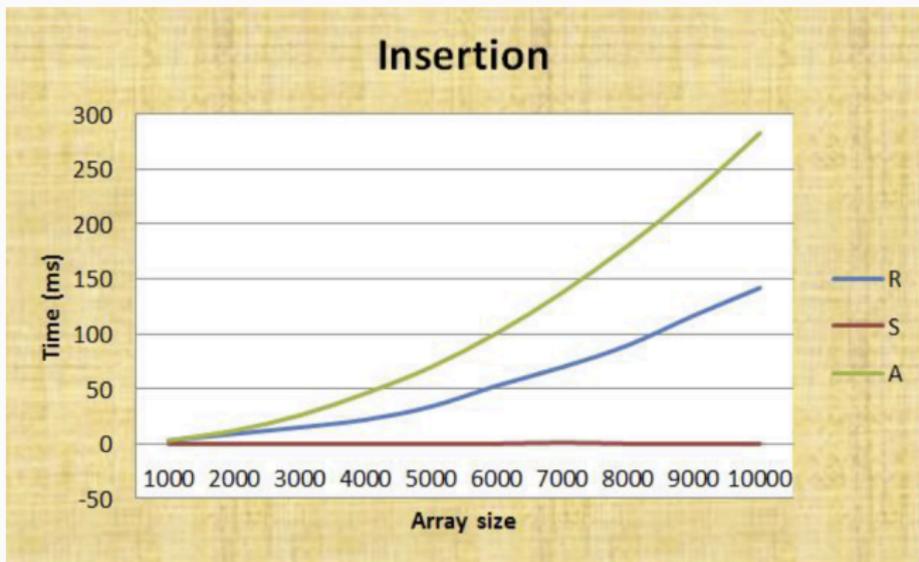
# Вычислительный эксперимент

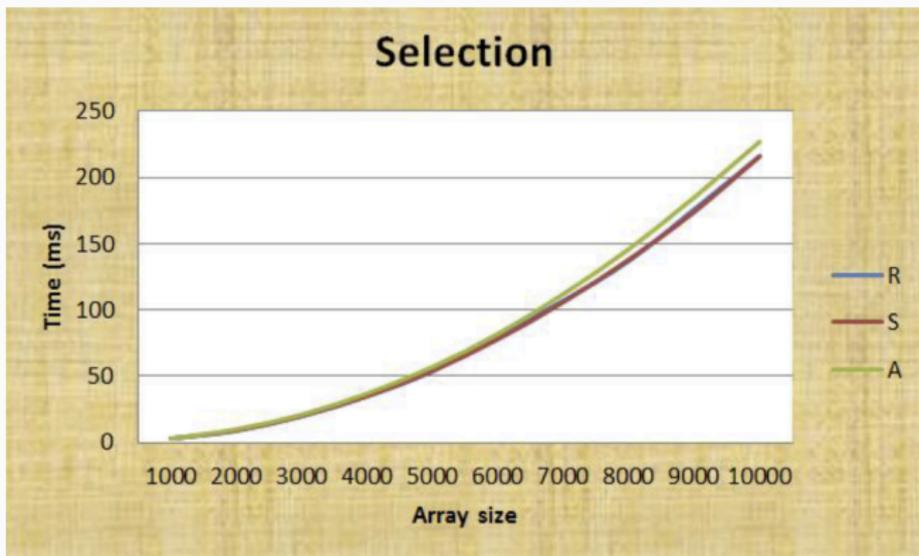
Теперь проведем вычислительный эксперимент и измерим время работы сортировок

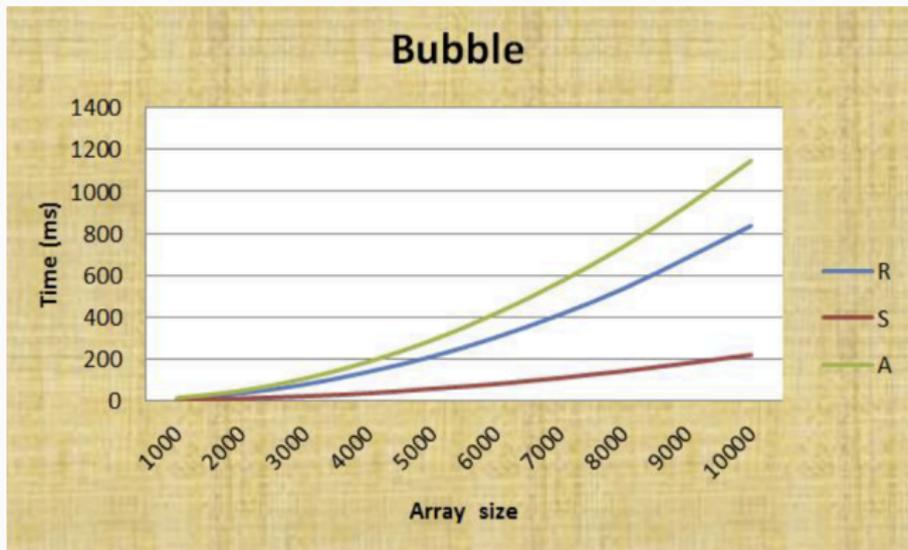
Insertion			
N	R	S	A
1000	3	0	3
2000	9	0	12
3000	15	0	26
4000	22	0	46
5000	34	0	70
6000	53	0	101
7000	70	1	138
8000	90	0	181
9000	117	0	229
10000	142	0	283

Selection			
N	R	S	A
1000	3	3	3
2000	9	9	10
3000	20	20	21
4000	35	35	37
5000	54	54	57
6000	81	78	82
7000	107	106	112
8000	137	138	146
9000	176	174	185
10000	216	216	227

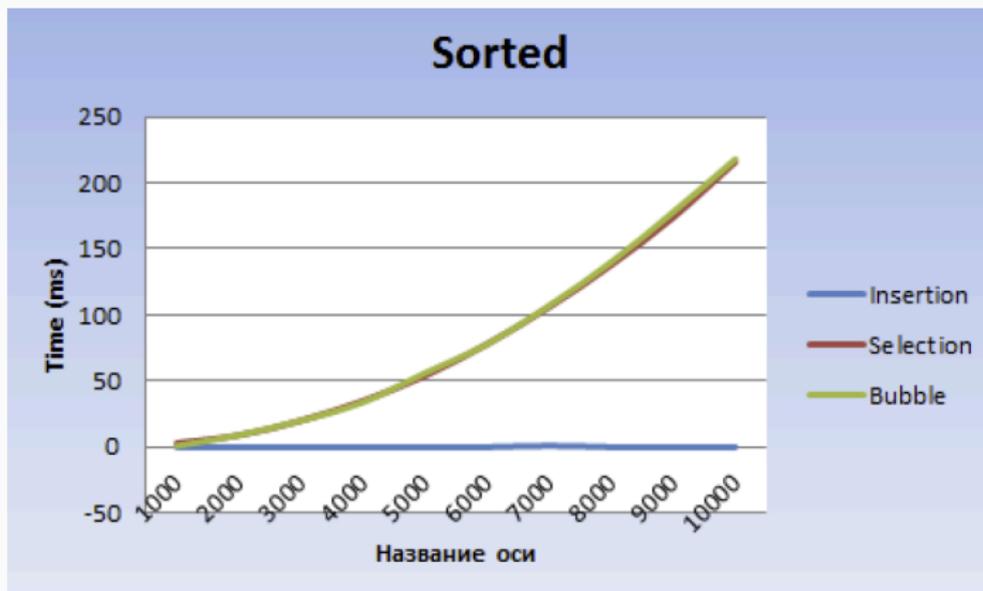
Bubble			
N	R	S	A
1000	7	1	12
2000	34	9	46
3000	76	20	104
4000	137	34	185
5000	210	56	288
6000	302	78	416
7000	410	107	566
8000	535	140	736
9000	683	178	932
10000	836	218	1148











## Список литературы

---

-  Кормен Т., Лейзерсон Ч., Ривест Р.  
*Алгоритмы: построение и анализ*  
МЦНМО, Москва, 2000
-  Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы:  
построение и анализ.  
*2-е изд. — М.: «Вильямс», 2006*
-  Википедия  
*Алгоритм*  
<http://ru.wikipedia.org/wiki/Алгоритм>
-  Википедия  
*Список алгоритмов*  
[http://ru.wikipedia.org/wiki/Список\\_алгоритмов](http://ru.wikipedia.org/wiki/Список_алгоритмов)
-  Традиция  
*Задача коммивояжёра*  
<http://traditio.ru/wiki/Задача>

-  Википедия  
*NP-полная задача*  
<http://ru.wikipedia.org/wiki/NP-полная>
-  Серджвик Р.  
*Фундаментальные алгоритмы на C++. Части 1-4*  
Diasoft, 2001
-  Седжвик Р.  
*Фундаментальные алгоритмы на C. Анализ/Структуры данных/Сортировка/Поиск*  
СПб.: ДиаСофтЮП, 2003
-  Седжвик Р.  
*Фундаментальные алгоритмы на C. Алгоритмы на графах*  
СПб.: ДиаСофтЮП, 2003
-  Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.  
*Издательский дом «Вильямс», 2000*

-  Кнут Д.  
*Искусство программирования, том 1. Основные алгоритмы*  
3-е изд. — М.: «Вильямс», 2006
-  Кнут Д.  
*Искусство программирования, том 2. Получисленные методы*  
3-е изд. — М.: «Вильямс», 2007
-  Кнут Д.  
*Искусство программирования, том 3. Сортировка и поиск*  
2-е изд. — М.: «Вильямс», 2007
-  Кнут Д.  
*Искусство программирования, том 4, выпуск 3. Генерация всех сочетаний и разбиений*  
М.: «Вильямс», 2007



Кнут Д.

*Искусство программирования, том 4, выпуск 4. Генерация всех деревьев. История комбинаторной генерации*

М.: «Вильямс», 2007