

# ВОПРОСЫ

**Что такое массив?**

**Что такое индекс элемента массива?**

**Какие действия можно производить с массивами?**

# СОРТИРОВКА МАССИВА

АЛГОРИТМЫ СОРТИРОВКИ

# СОРТИРОВКА -

(англ. *sorting* — классификация, упорядочение) — последовательное расположение или разбиение на группы чего-либо в зависимости от выбранного критерия.

Алгоритм сортировки — это алгоритм для упорядочивания элементов в списке. В случае, когда элемент списка имеет несколько полей, поле, служащее критерием порядка, называется ключом сортировки. На практике в качестве ключа часто выступает число, а в остальных полях хранятся какие-либо данные, никак не влияющие на работу алгоритма.

Первые прототипы современных методов сортировки появились уже в XIX веке. К 1890 году для ускорения обработки данных переписи населения в США американец Герман Холлерит создал первый статистический табулятор — электромеханическую машину, предназначенную для автоматической обработки информации, записанной на перфокартах<sup>[1]</sup>. У машины Холлерита имелся специальный «сортировальный ящик» из 26 внутренних отделений.

В дальнейшем история алгоритмов оказалась связана с развитием электронно-вычислительных машин. По некоторым источникам, именно программа сортировки стала первой программой для вычислительных машин.

## ИСТОРИЯ



# ОЦЕНКА АЛГОРИТМА СОРТИРОВКИ

Время — основной параметр, характеризующий быстродействие алгоритма. Называется также вычислительной сложностью.

Память — ряд алгоритмов требует выделения дополнительной памяти под временное хранение данных.

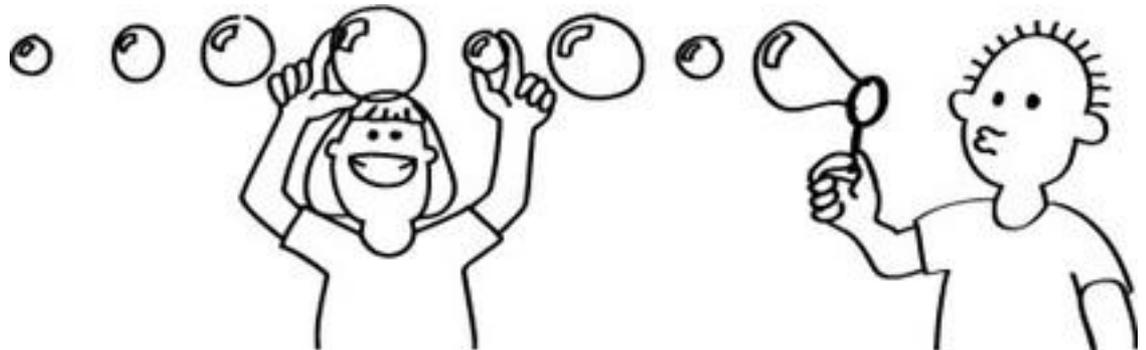
# АЛГОРИТМЫ СОРТИРОВКИ

- 1.** Сортировка пузырьком
- 2.** Сортировка вставками
- 3.** Гномья сортировка
- 4.** Быстрая сортировка
- 5.** Сортировка Шелла

# СОРТИРОВКА ПРОСТЫМИ ОБМЕНАМИ или СОРТИРÓВКА ПУЗЫРЬКÓМ

(англ. *bubble sort*) — простой алгоритм сортировки. Для понимания и реализации этот алгоритм — простейший, но эффективен он лишь для небольших массивов.

Алгоритм считается учебным и практически не применяется вне учебной литературы, вместо него на практике применяются более эффективные алгоритмы сортировки. В то же время метод сортировки обменами лежит в основе некоторых более совершенных алгоритмов, таких как шейкерная сортировка, пирамидальная сортировка и быстрая сортировка.



# СОРТИРОВКА ПРОСТЫМИ ОБМЕНАМИ ИЛИ СОРТИРÓВКА ПУЗЫРЬКÓМ

```
for i := 1 to m-1 do
for j := 1 to m-i do
if arr[j] > arr[j+1] then
begin
k := arr[j];
arr[j] := arr[j+1];
arr[j+1] := k
end;
```

6 5 3 1 8 7 2 4

# СОРТИРОВКА ВСТАВКАМИ

**(англ. *Insertion sort*) — алгоритм сортировки, в котором элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов**

# СОРТИРОВКА ВСТАВКАМИ

```
begin
  for i:=2 to N do
    begin
      buf:=x[i];
      j:=i-1;
      while (j>=1) and (x[j]>buf) do
        begin
          x[j+1]:=x[j];
          j:=j-1;
        end;
      x[j+1]:=buf;
    end;
  end;
end;
```

6 5 3 1 8 7 2 4

Алгоритм сортировки, похожий на сортировку вставками, но в отличие от последней перед вставкой на нужное место происходит серия обменов, как в сортировке пузырьком. Название происходит от предполагаемого поведения садовых гномов при сортировке линии садовых горшков.

*« Гномья сортировка основана на технике, используемой обычным голландским садовым гномом (нидерл. *tuinkabouter*). Это метод, которым садовый гном сортирует линию цветочных горшков. По существу он смотрит на текущий и предыдущий садовые горшки: если они в правильном порядке, он шагает на один горшок вперёд, иначе он меняет их местами и шагает на один горшок назад. Граничные условия: если нет предыдущего горшка, он шагает вперёд; если нет следующего горшка, он закончил.»*

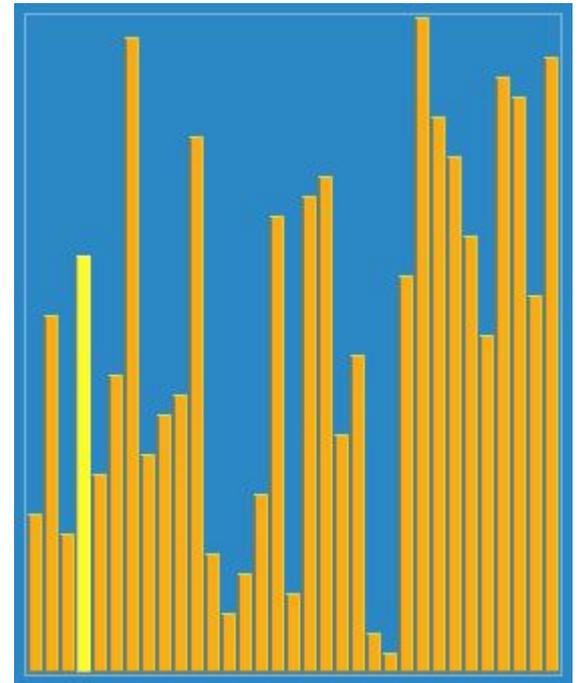
Дик Грун



ГНОМЬЯ СОРТИРОВКА

```
begin
i := 2;
j := 3;
while i <= size do
begin
if arr[i-1] <= arr[i] then
begin
i := j;
j := j + 1
end
else
begin
t := arr[i-1];
arr[i-1] := arr[i];
arr[i] := t;
i := i - 1;
if i = 1 then
begin
i := j;
j := j + 1
end
end
end;
end;
```

# ГНОМЬЯ СОРТИРОВКА



# БЫСТРАЯ СОРТИРОВКА

**Быстрая сортировка, сортировка Хоара** ([англ. quicksort](#)), часто называемая **qsort** (по имени в стандартной [библиотеке](#) языка [Си](#)) — широко известный [алгоритм сортировки](#), разработанный английским информатиком [Чарльзом Хоаром](#) во время его работы в [МГУ](#) в [1960 году](#).

**Общая идея алгоритма состоит в следующем:**

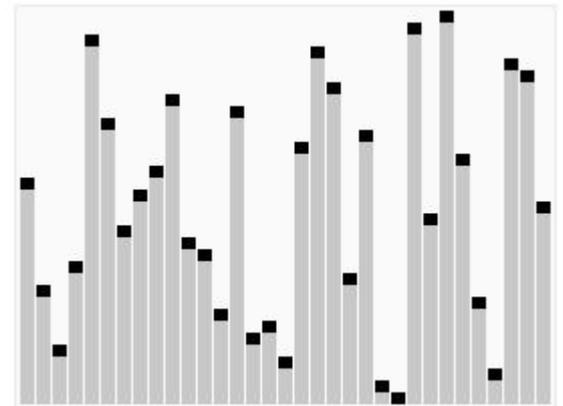
Выбрать из массива элемент, называемый опорным. Это может быть любой из элементов массива или же число, вычисленное на основе значений элементов.

Сравнить все остальные элементы с опорным и переставить их в массиве так, чтобы разбить массив на три непрерывных отрезка, следующие друг за другом: «меньшие опорного», «равные» и «большие». <sup>[1]</sup>

Для отрезков «меньших» и «больших» значений выполнить рекурсивно ту же последовательность операций, если длина отрезка больше единицы.

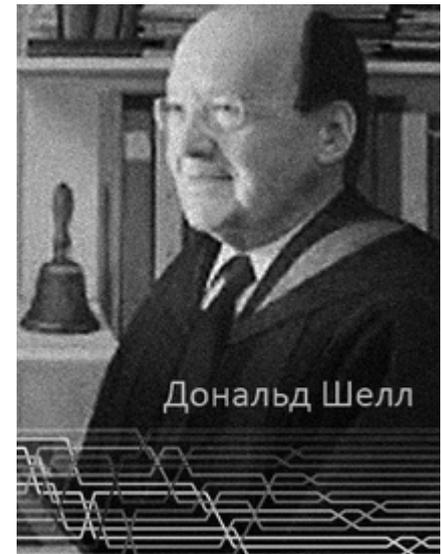
# БЫСТРАЯ СОРТИРОВКА

```
begin
  i:=l;
  j:=r;
  m:=round ((l+r)/2);{средний элемент}
  x1:=x[m];
  repeat
    while x[i]>x1 do inc(i);{пока левый больше среднего, подвигаем левый
край вправо }
    while x[j]<x1 do dec(j);{пока правый меньше среднего, подвигаем левый
вправо}
    if i<=j then {если левый и правый срослись}
      begin
        y1:=x[i];
        x[i]:=x[j];{меняем левый и правый}
        x[j]:=y1;
        inc(i); {левый вправо}
        dec(j); {правый влево}
      end;
    until i>j;{конец одной перестановки}
    if l<j then sort(l,j);{рекурсивно сортируем}
    if i<r then sort(i,r);{или левую или правую части}
  end;
```



# СОРТИРОВКА ШЕЛЛА

(англ. *Shell sort*) — алгоритм сортировки, являющийся усовершенствованным вариантом сортировки вставками. Идея метода Дональда Шелла состоит в сравнении элементов, стоящих не только рядом, но и на определённом расстоянии друг от друга; иными словами — это сортировка вставками, но с предварительными «грубыми» проходами.



# СОРТИРОВКА ШЕЛЛА

```
procedure ShellSort( var A : mas );  
  const  
    steps = 12;  
  var  
    i, j, l, k, p, n : Integer;  
    x : itp;  
    s : array [1..steps] of Integer;  
  begin  
    k := 1;  
    { Формируем последовательность чисел -  
      шаги, с которыми выбираем сортируемые  
      подмассивы }  
    for i := steps downto 1 do  
      begin  
        s[i] := k;  
        k := k * 2 + 1;  
      end;
```

