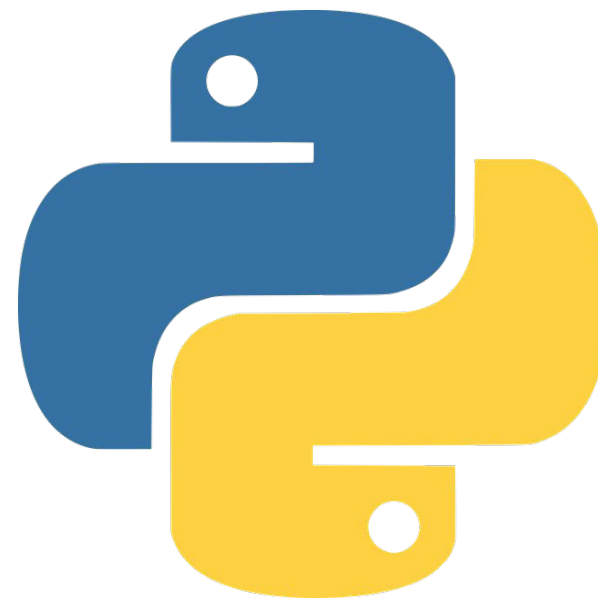


# Сортировка в массивах



# Методы сортировки

## Простые

*Вставками*

*Подсчетом  
Выбором*

*Обменом*

## Сложные

*Слияниями*

*Пирамидальная*

*С  
разделением  
Метод*

*Шелла*

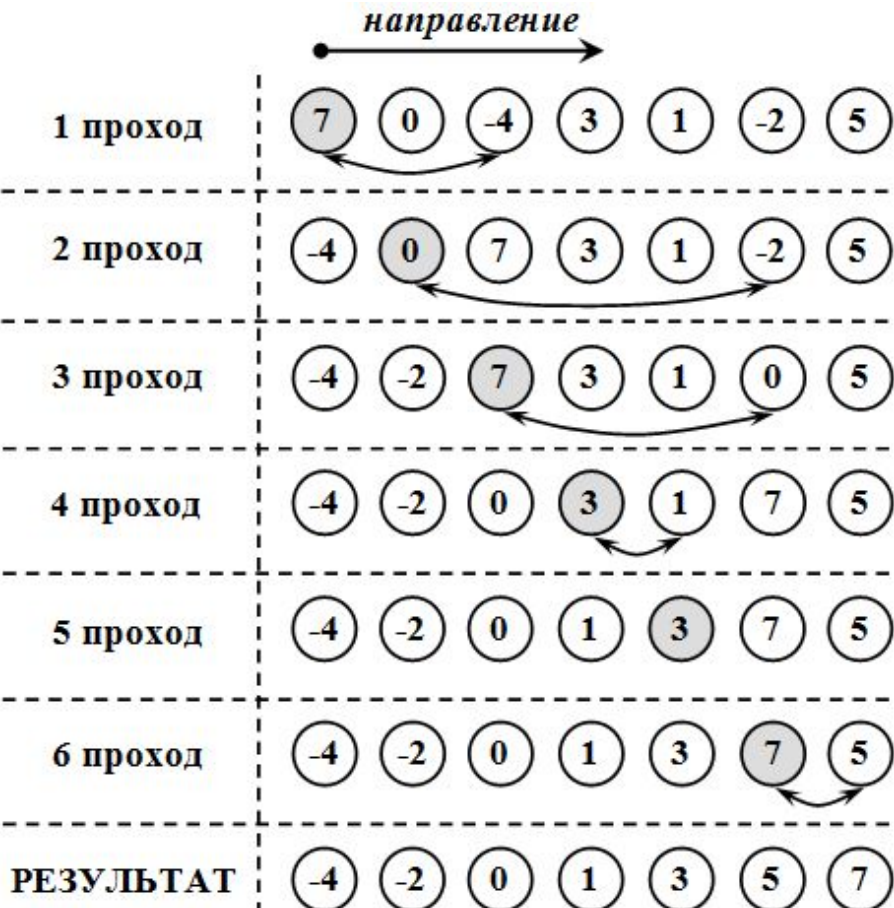
## Рассматриваемые

*Сортировка вставками*

*Сортировка*

*выбором  
Быстрая сортировка*

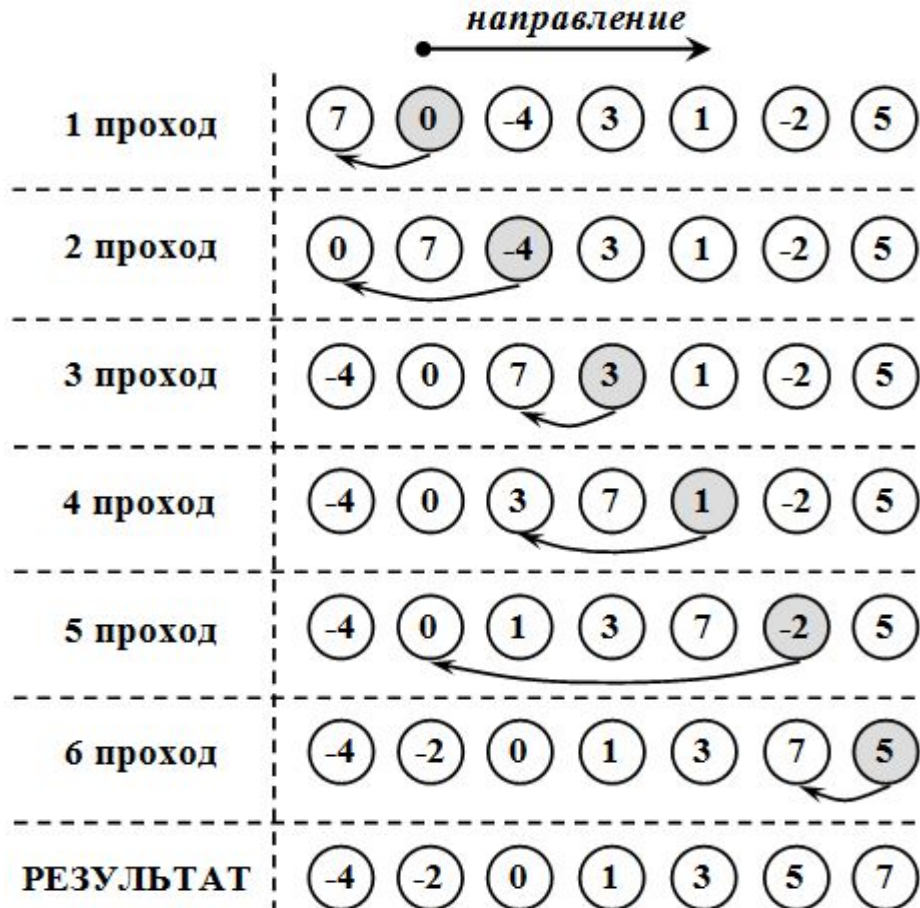
# Сортировка выбором



## Демонстрация

```
void SelectionSort (int k,int x[max]) {  
    int i,j,min,temp;  
    for (i=0;i<k-1;i++) {  
        //устанавливаем начальное значение минимально-  
        //минимума  
        min=i; // -го индекса  
        //находим минимальный индекс элемента  
        for (j=i+1;j<k;j++){  
            if (x[j]<x[min])  
                min=j;  
        }  
        //меняем значения местами  
        temp=x[i];  
        x[i]=x[min];  
        x[min]=temp; }  
}
```

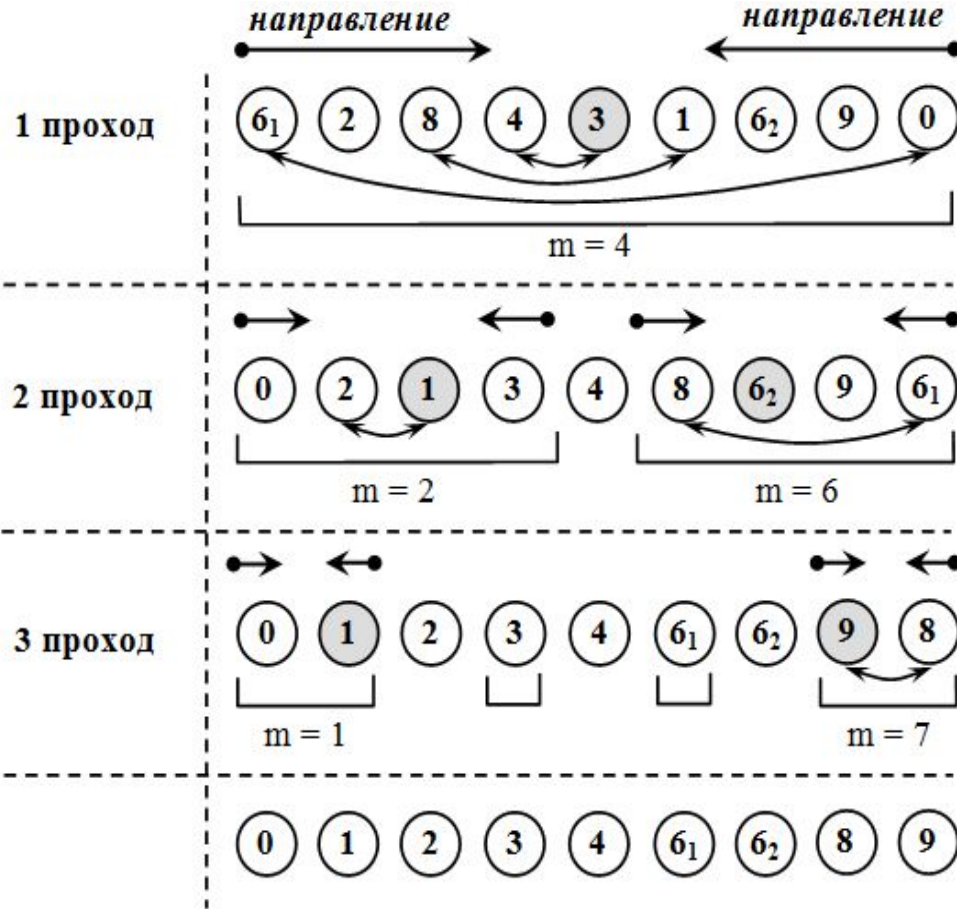
# Сортировка вставкой



## Демонстрация

```
void InsertSort (int k,int x[max]) {
  int i,j, temp;
  for (i=0;i<k;i++) {
    //цикл проходов, i - номер прохода
    temp=x[i];
    //поиск места элемента
    for (j=i-1; j>=0 && x[j]>temp; j--)
      x[j+1]=x[j];/*сдвигаем элемент вправо, пока
                  не дошли */
    // место найдено, вставить элемент
    x[j+1]=temp; }
}
```

# Быстрая сортировка Хоара



## Демонстрация

```
void Hoar_Sort (int k, int *x){
    Quick_Sort (0, k-1, x);}
void Quick_Sort(int left, int right, int *x){
    int i, j, m, h;
    i = left;
    j = right;
    m = x[(i+j+1)/2];
    do {
        while (x[i] < m) i++;
        while (x[j] > m) j--;
        if (i <= j) {
            Exchange(i,j,x);
            i++;
            j--; }
    } while(i <= j);
    if (left < j)
        Quick_Sort (left, j, x);
    if (i < right)
        Quick_Sort (i, right, x);}
//процедура обмена двух элементов
void Exchange (int i, int j, int *x){
    int tmp;
    tmp = x[i];
    x[i] = x[j];
    x[j] = tmp; }
```