

# Различные методы сортировки

Занятие 1

# 1) Сортировка Выбором (Selection-sort)

- берем первый элемент последовательности  $A[i]$ ;
- находим минимальный (максимальный) элемент последовательности и запоминаем его номер;
- если номер первого элемента и номер найденного элемента не совпадают, тогда два этих элемента обмениваются значениями, иначе никаких манипуляций не происходит;
- увеличиваем  $i$  на 1 и продолжаем сортировку оставшейся части массива, а именно с элемента с номером 2 по  $N$ , так как элемент  $A[1]$  уже занимает свою позицию;

## 2) сортировка пузырьком (*bubble sort*)

- пузырек воздуха в стакане воды поднимается со дна вверх.

Для массивов – **самый маленький** («легкий» элемент перемещается вверх («**всплывает**»)).

- сравниваем два соседних элемента; если они стоят «неправильно», меняем их местами
- за 1 проход по массиву **один** элемент (самый маленький) становится на свое место

# Сортировка Шейкерная- Перемешиванием (Shaker, Cocktail sort)

- двунаправленность: алгоритм перемещается, ни как в обменной (пузырьковой) сортировке – строго снизу вверх (слева направо), а сначала снизу вверх, потом сверху вниз.

### 3) Сортировка подсчётом (*counting sort*)

достаточно завести массив и хранить в нем количество повторений каждого целого числа в массиве, а затем последовательно пробежаться по массиву и вывести каждое число столько раз, сколько указано в массиве.

# 4) Поразрядная сортировка

## RadixSort

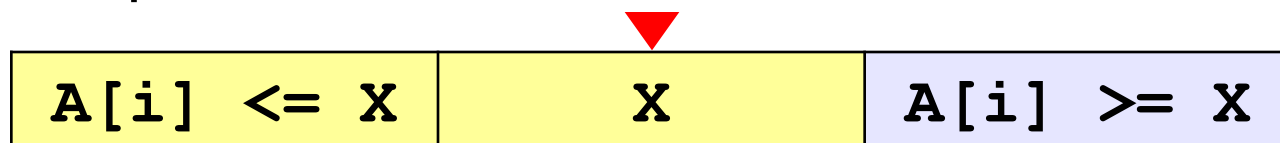
# 5) Быстрая сортировка QuickSort

- разбиение массива относительно опорного элемента;
- рекурсивная сортировка каждой части массива.

# Быстрая сортировка

**Шаг 1:** выбрать некоторый элемент массива  $X$

**Шаг 2:** переставить элементы так:



**Шаг 3:** если в подмассиве более двух элементов, рекурсивно запускаем для него ту же процедуру.

Разделяй и властвуй (англ. *divide and conquer*)

78	6	82	67	55	44	34
----	---	----	----	----	----	----

Продолжая деление этих половин до тех пор пока не останется в них по 1 элементу.



Как лучше выбрать  $X$ ?

**Выбираем средний элемент массива.**



# Быстрая сортировка, разбиение массива

---

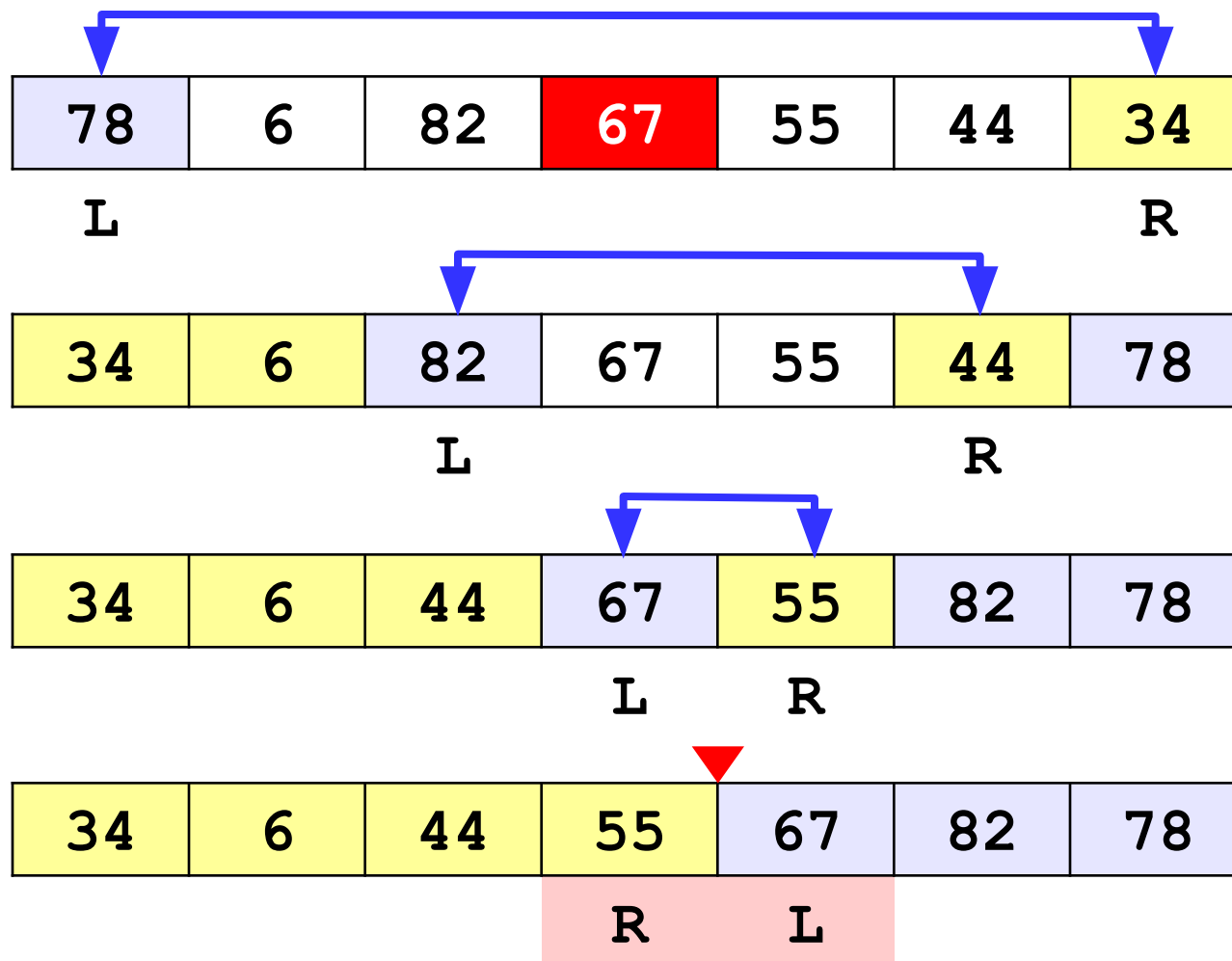
## Разделение:

- 1) выбрать средний элемент массива ( $x=67$ )

78	6	82	67	55	44	34
----	---	----	----	----	----	----

- 2) установить  $L := 1, R := N$
- 3) увеличивая  $L$ , найти первый элемент  $A[L]$ , который  $\geq x$  (должен стоять справа)
- 4) уменьшая  $R$ , найти первый элемент  $A[R]$ , который  $\leq x$  (должен стоять слева)
- 5) если  $L \leq R$  то поменять местами  $A[L]$  и  $A[R]$  и перейти к п. 3  
иначе **СТОП**.

# Быстрая сортировка



**L > R : разделение закончено!**

# БЫСТРАЯ СОРТИРОВКА

---

```
program a_1;
```

Описание переменных и массива

```
procedure
```

процедура быстрой сортировки и

рекурсивные ее вызовы для правой и левой части

```
begin
```

Заполнение исходного массива и его вывод

```
qSort(n,m); обращение к процедуре
```

```
for i:=1 to 10 do write(a[i], ' '); вывод
```

отсортированного массива

```
end.
```

# БЫСТРАЯ СОРТИРОВКА

---

```
procedure qSort(nStart, nEnd: integer);
var L, R, c, X: integer;
begin
  if nStart >= nEnd then Exit;
  L:= nStart; R:= nEnd;
  X:= A[(L+R) div 2];
  while L <= R do begin
    while A[L] < X do L:= L + 1;
    while A[R] > X do R:= R - 1;
    if L <= R then begin
      c:= A[L]; A[L]:= A[R]; A[R]:= c;
      L:= L+1; R:= R-1
    end;
  end;
  qSort(nStart, R);
  qSort(L, nEnd)
end;
```

# Быстрая сортировка ХОАРА

---

## 6) Сортировка слиянием MergeSort

- Сортировка слиянием - это рекурсивный алгоритм. Он, также как и быстрая сортировка(описано в первой части), делит список на две части, и затем рекурсивно вызывает сам себя для их дальнейшего упорядочивания. Она делит список на две равные части, после чего подсписки рекурсивно сортируются и сливаются для того чтобы образовать полностью отсортированный список.