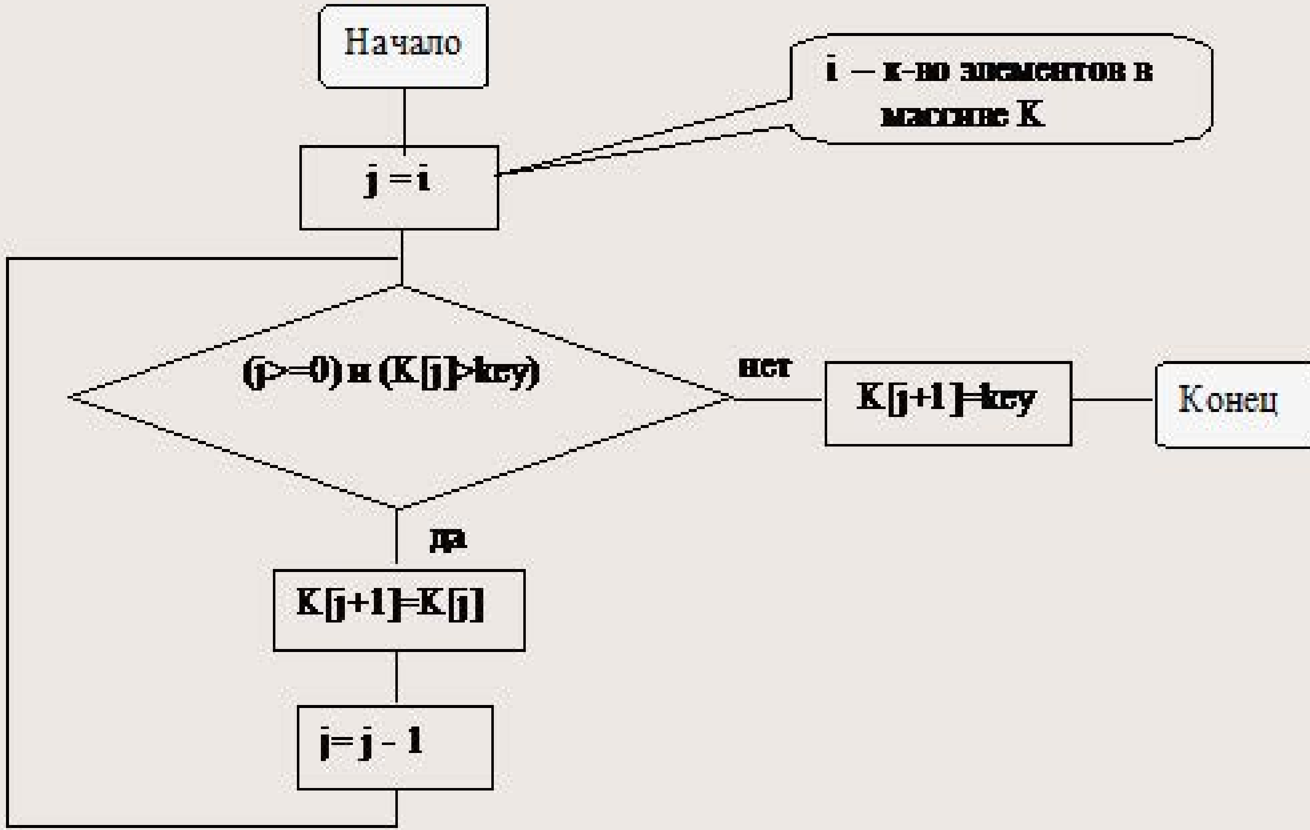
6. Сортировка…

Определение 3.4. Действия, связанные с размещением записей в порядке возрастания (или убывания) ключей называют *сортировкой*

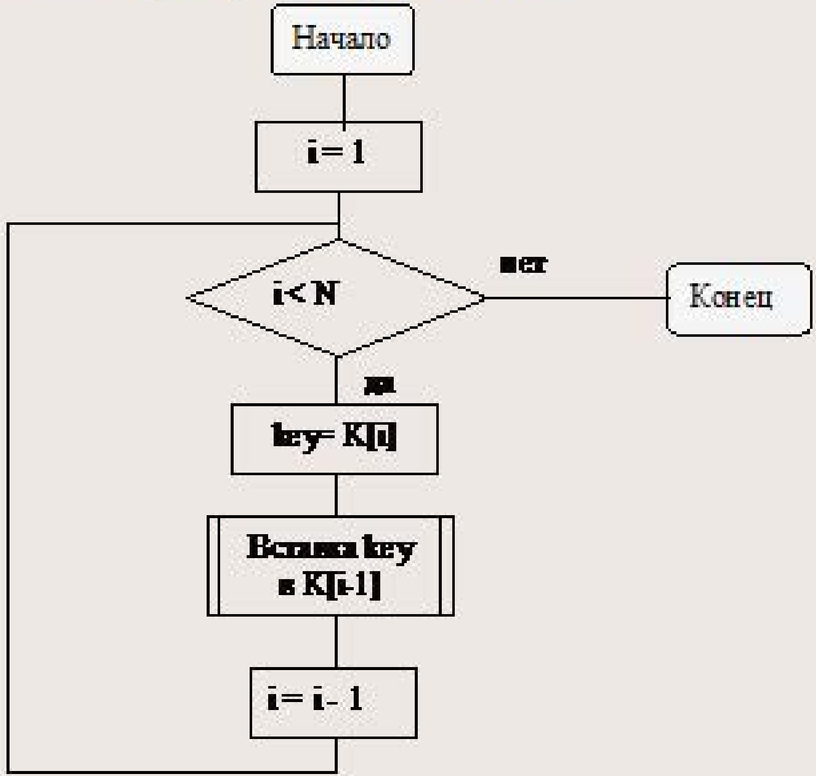
Определение 3.5. Алгоритм сортировки называют *устойчивым* , если он никогда не меняет относительный порядок в таблице двух записей с равными ключами

Определение 3.6. Упорядочивание данных, при которой все значения располагаются в ОП, называются *внутренней сортировкой* (проблема внешней сортировки остаются за рамками рассмотрения)

Идея похода – вставка нового значения в упорядоченный набор данных



В ходе сортировки –упорядочиваемый набор данных разбивается на две части – упорядоченный (слева) и неупорядоченный (справа) блоки

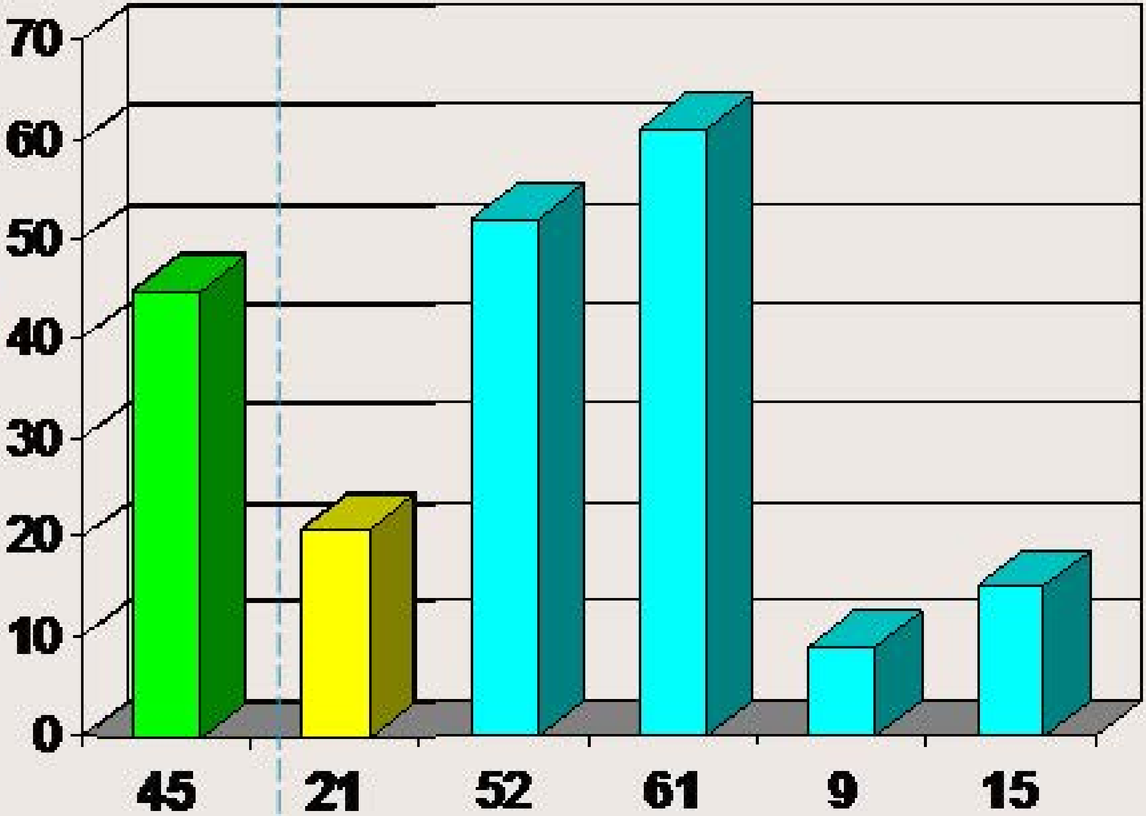


- Упорядоченная часть массива состоит из одного элемента

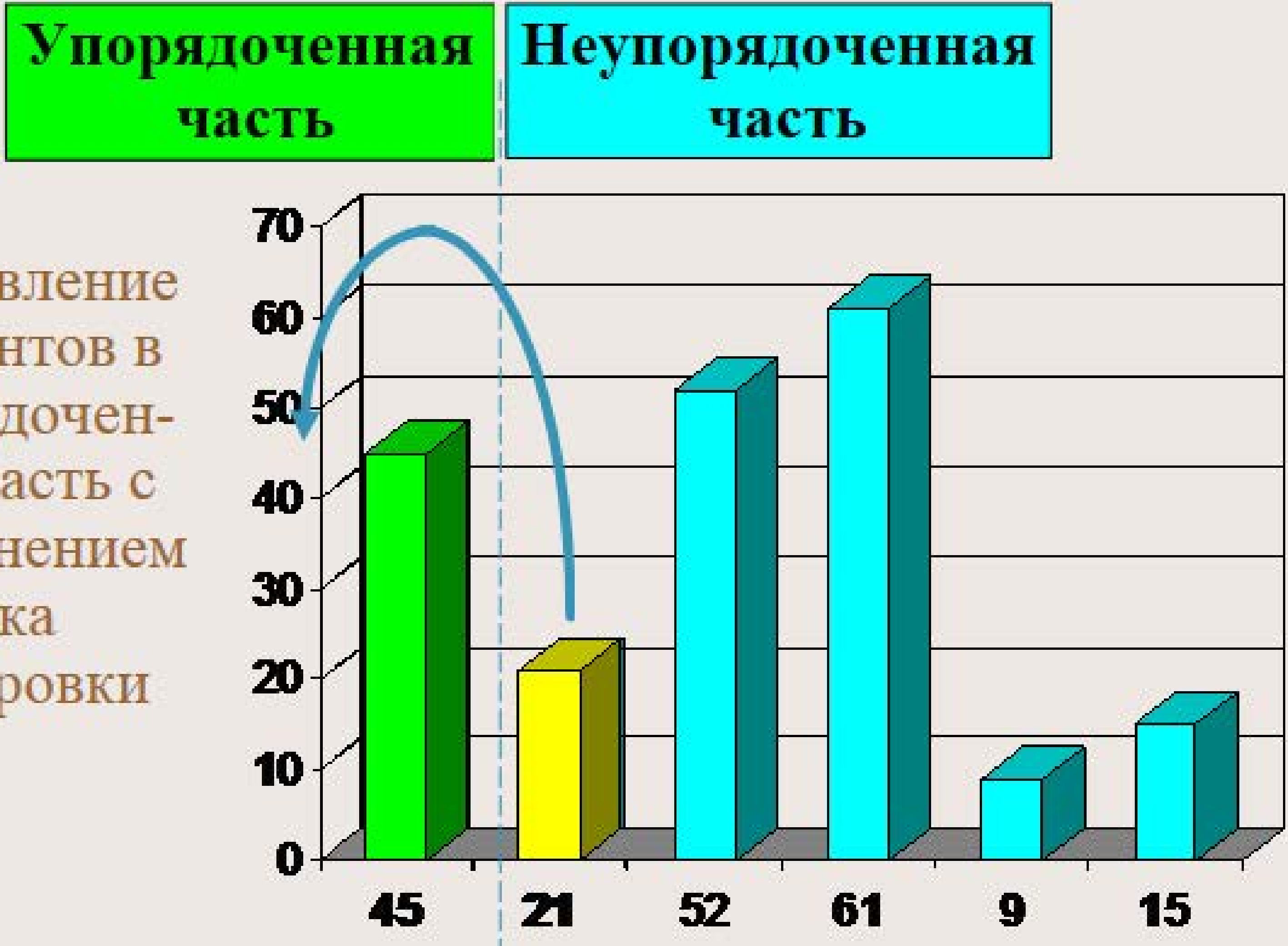
Упорядоченная часть

Неупорядоченная часть

- Упорядочен-ная часть массива увеличивается за счет первого элемента из неотсортиро-ванной части

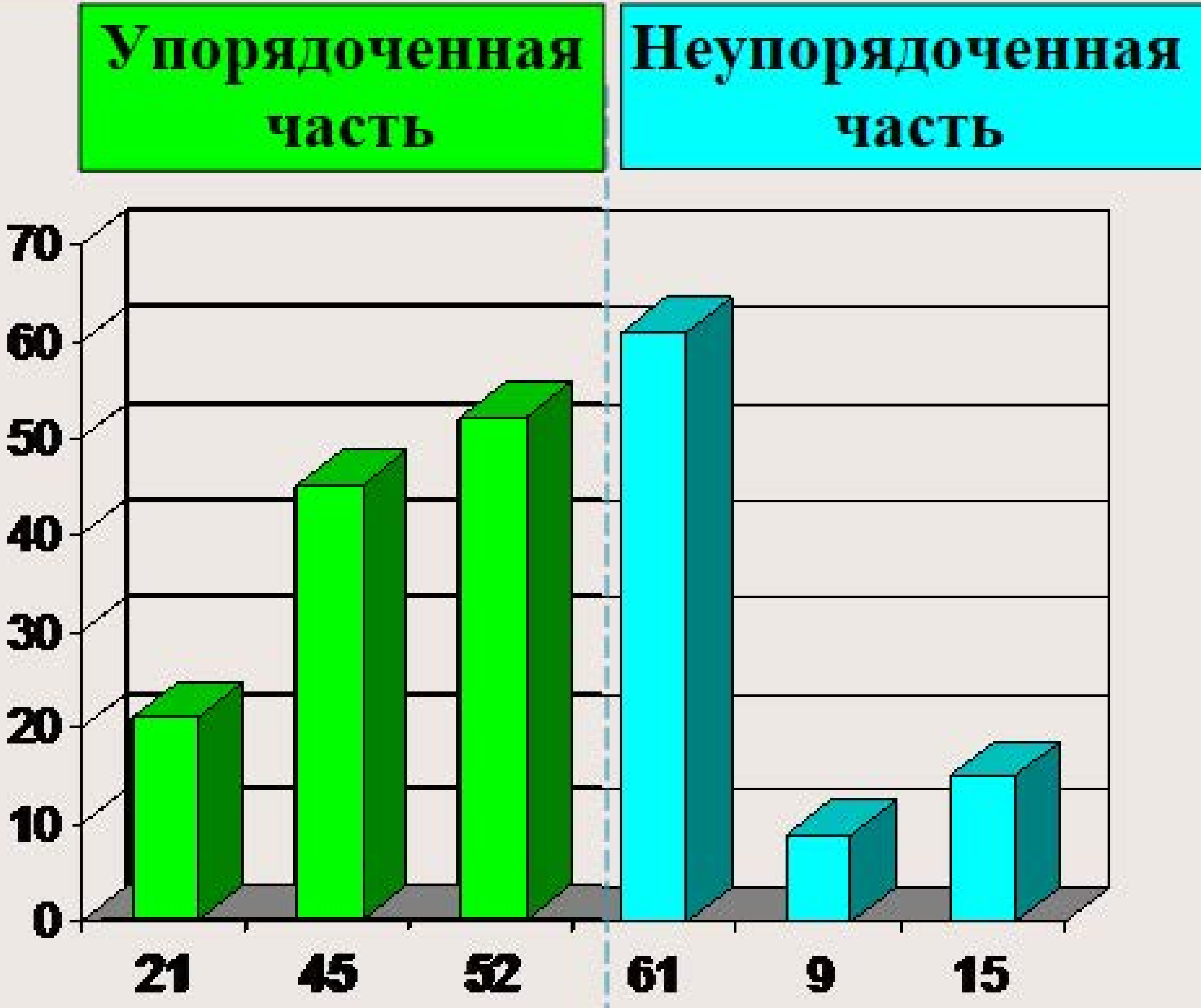


Добавление элементов в упорядочен-ную часть с сохранением порядка сортировки

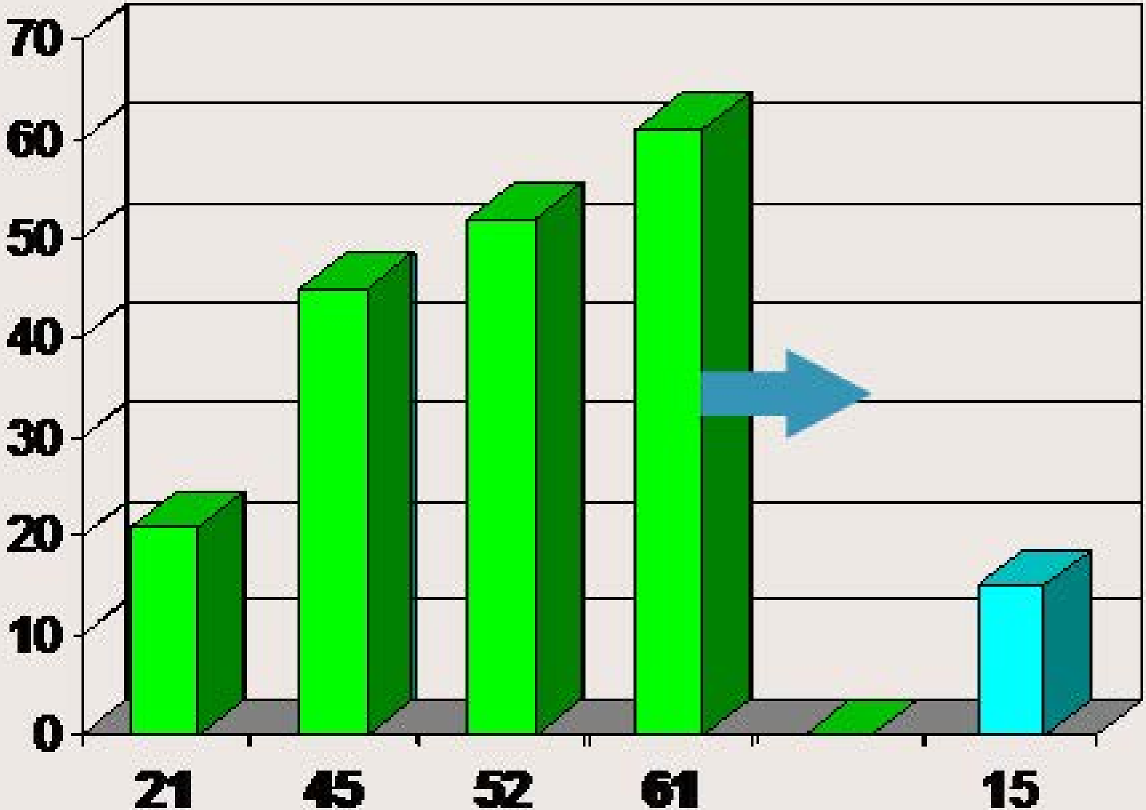


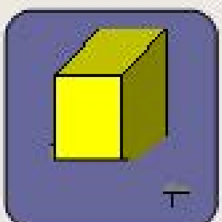
- Для сохранения порядка необходимо выполнить перепаковку (сдвиг) упорядоченной части

В ряде ситуаций перепаковка может отсутствовать

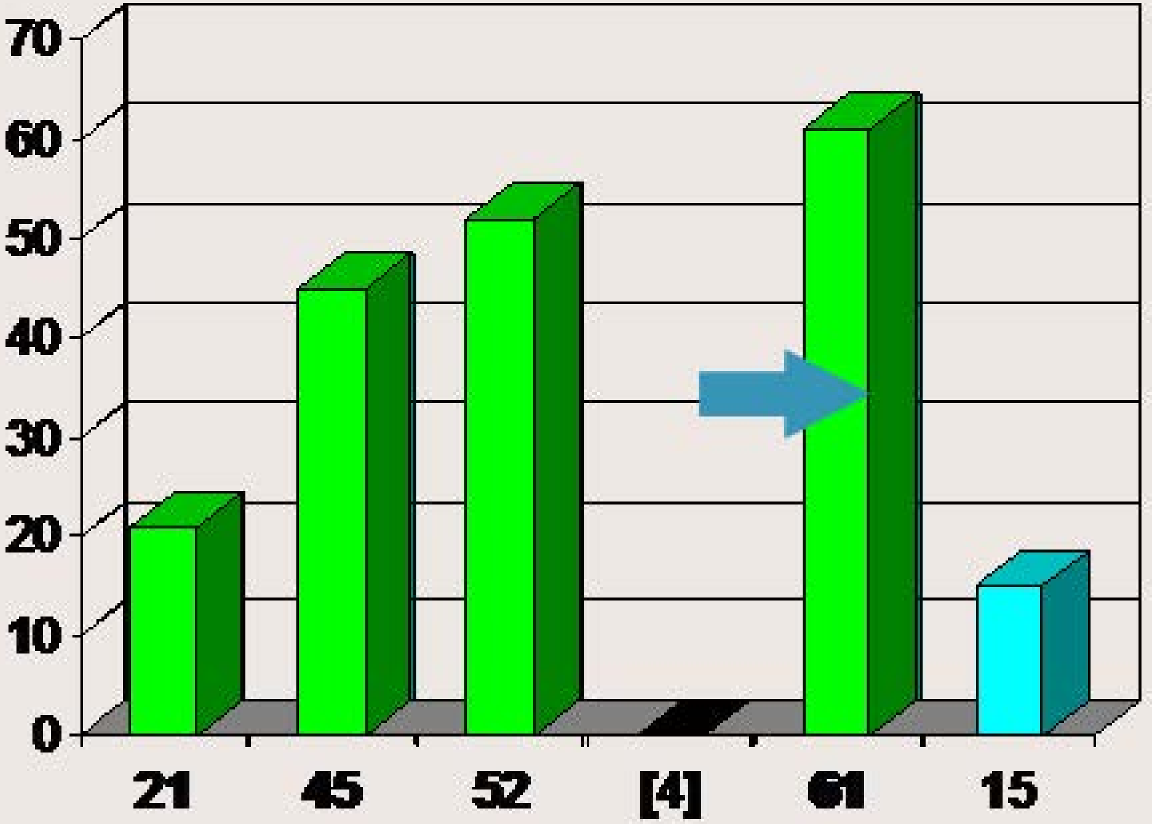


Вставляемый элемент копируется в отдельную переменную





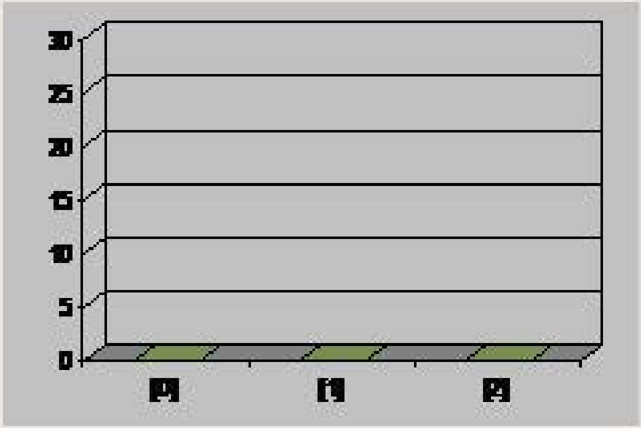
Далее выполняется последова-тельный сдвиг элементов

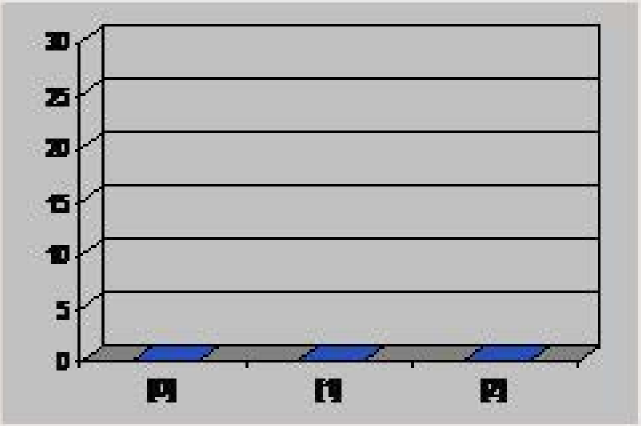


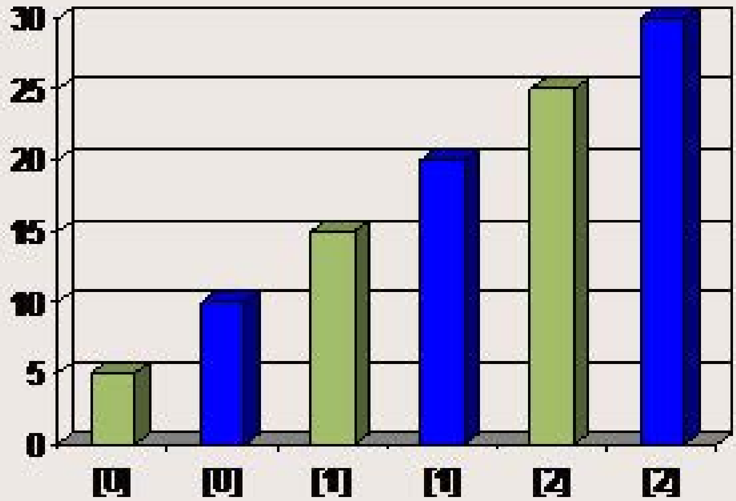
Оценка сложности

- $T_{min} = N$ (при сортировке упорядоченного массива)
- $T_{max} = N^2$ (= $1+2+\ldots+(N-1)$ – для каких данных ?)
- $T_{ср} = ?$

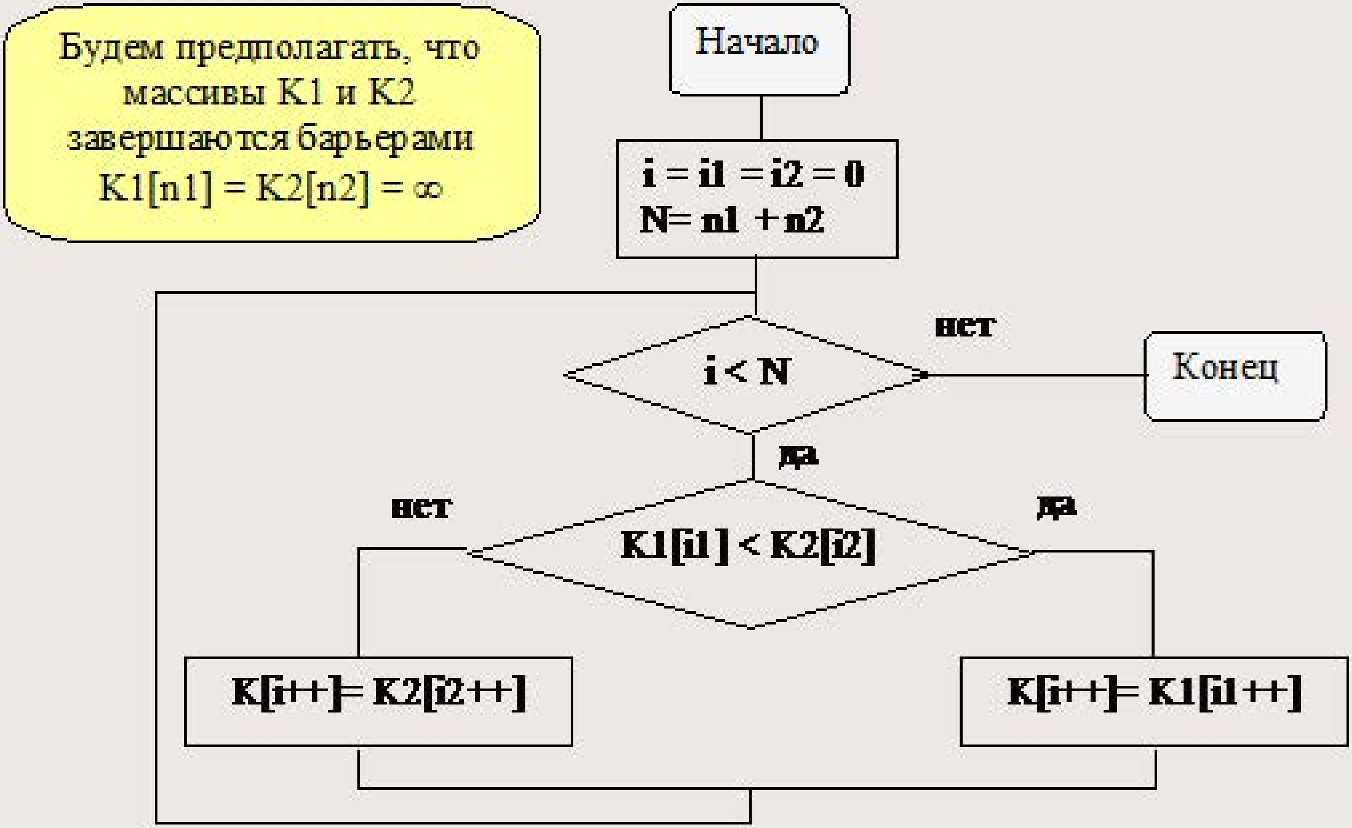
Идея похода – возможность эффективного объединения упорядоченных наборов данных

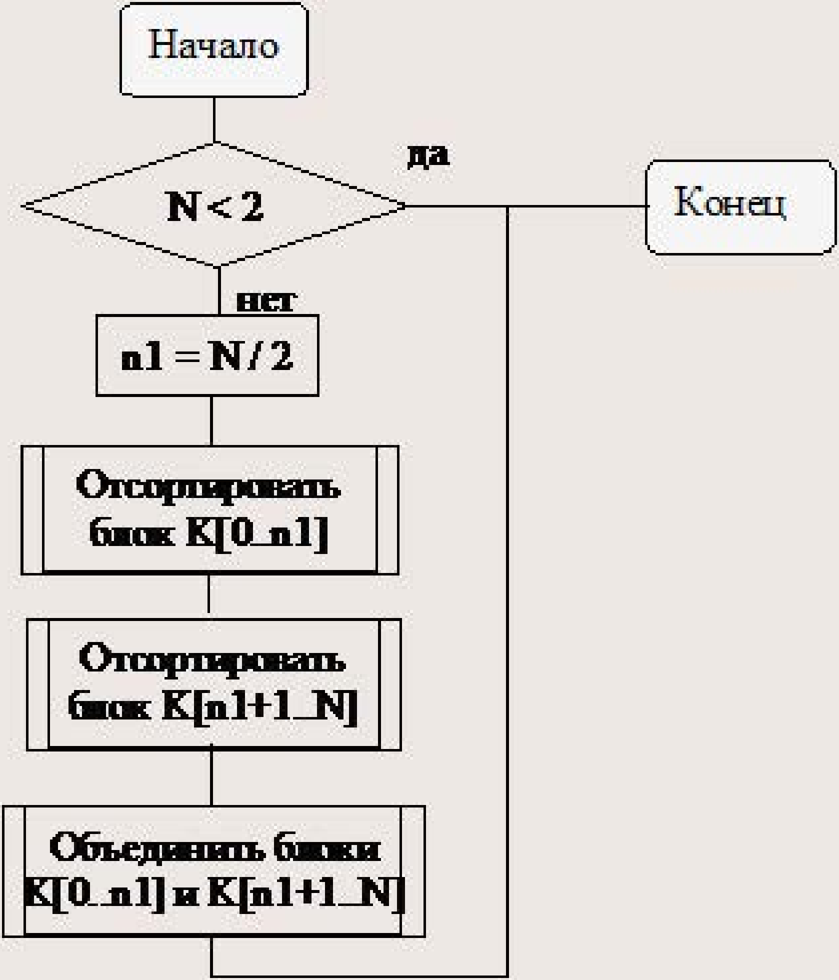






Слияние упорядоченных массивов




Начало
N < 2
да
Конец
нет
n1 = N / 2
Отсортировать блок K[0_n1]
Отсортировать блок K[n1+1_N]
Объединить блоки K[0_n1] и K[n1+1_N]

Оценка сложности

- ☑ N – количество сравнений при слиянии
- ☑ $\log_2 N$ – количество уровней рекурсии
- $T_{min} = T_{max} = T_{ср} = N \log_2 N$

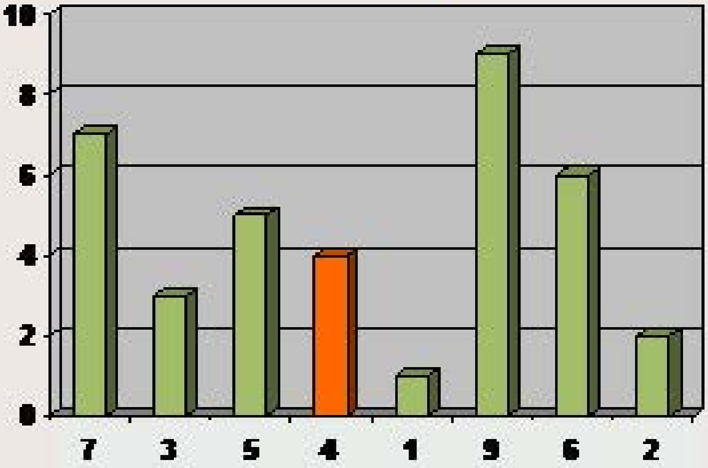
Реализация:

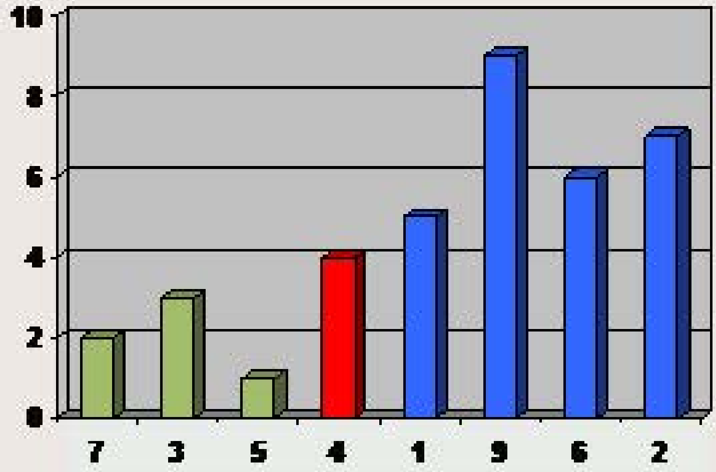
Экспериментальная оценка сложности:
программа, приложение

- Процедура слияния использует дополнительную память. **Пространственная сложность алгоритма (сложность по памяти)**

$$V = N$$

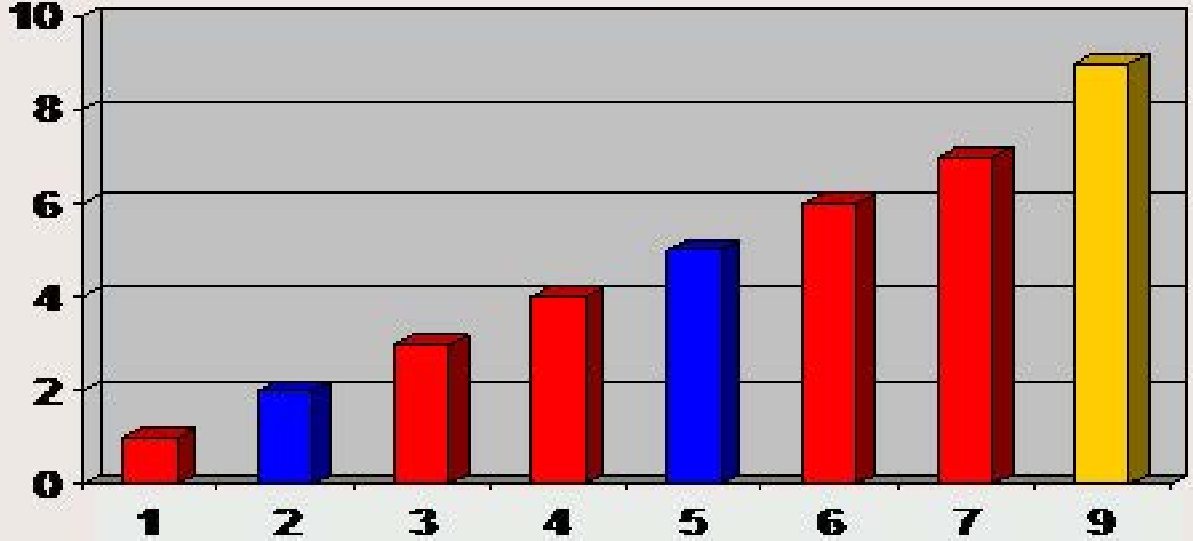
Идея похода (*Hoare C.A.R.*)– использование процедуры разделения упорядочиваемого набора на две части, в одной из которых располагаются значения, меньшие некоторого порогового (*ведущего*) элемента массива, в другой – соответственно большие значения. Подобный способ разделения может быть выполнен без привлечения дополнительной памяти.

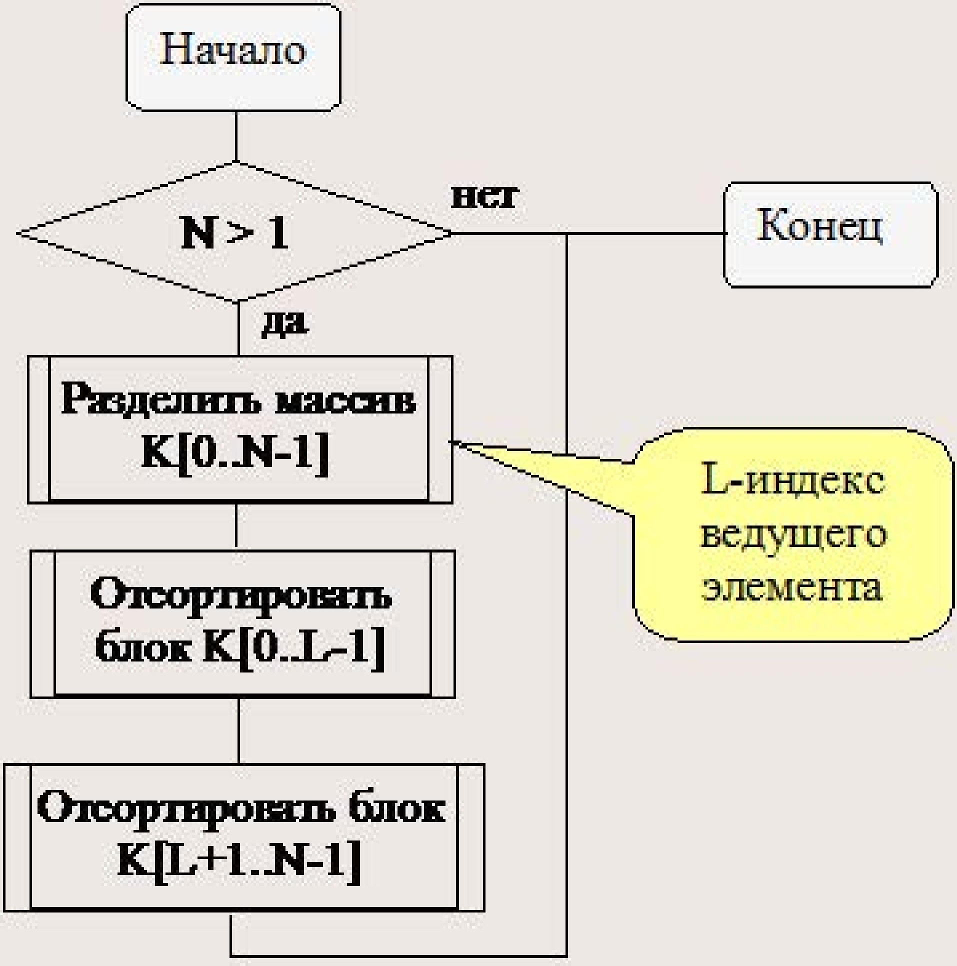




```
//  Разделение массива с использованием ведущего элемента
key = K[0];    // ведущий элемента
i1=1; i2=N-1; // индексы левого (i1) и правого (i2) блоков
// цикл, пока разделяемые блоки не пересекутся
while ( i1 <= i2 ) {
  //  пока K[i1]  не превышает вед. эл-т, переход вправо
  while ( (i1<N) && (K[i1] <= key) ) i1++;
  //  пока K[i2]  меньше вед. эл-та, переход влево
  while ( K[i2] > key ) i2--;
  // перестановка значений, которые
  // приостановили  разделение массива
  if ( i1 < i2 ) { kt = K[i1]; K[i1] = K[i2]; K[i2] = kt; }
}
//  установка ведущего элемента между блоками
K[0] = K[i2]; K[i2] = key;
i    = i2;  //  индекс ведущего элемента
```

При наличии процедуры разделения **алгоритм сортировки** может быть определен *рекурсивно* – необходимо <u>разбить</u> упорядочиваемый набор на два блока с меньшими и большими значениями соответственно и затем последовательно <u>отсортировать</u> полученные блоки




Начало
N > 1
нет
Конец
да
Разделить массив K[0..N-1]
L-индекс ведущего элемента
Отсортировать блок K[0..L-1]
Отсортировать блок K[L+1..N-1]

Оценка сложности

- $T_{min} = N \log_2 N$
- $T_{max} = N^2$ (!!!)
- $T_{ср} = ?$

Вероятность выбора любого ключа в качестве ведущего элемента является одинаковой

$$T(N) \le \alpha N + \frac{1}{N}\sum_{i=1}^{N-1}[T(i-1)+T(N-i)] = \alpha N + \frac{2}{N}\sum_{i=0}^{N-1}T(i)$$

Докажем по индукции, что

$$T(N) \le KN \ln N \; (K = 2\alpha + 2\beta,\ \beta = T(0) = T(1))$$

$$\Rightarrow T(N) \le \alpha N + \frac{2}{N}(T(0)+T(1)) + \frac{2}{N}\sum_{i=2}^{N-1} Ki \ln i$$

$$T\,к.\ \frac{2}{N}\sum_{i=2}^{N-1} Ki\ln i \le \int_2^N x\ln x\,dx \le \frac{N^2\ln N}{2}-\frac{N^2}{4}$$

$$\Rightarrow T(N)\quad \le \alpha N+\frac{4\beta}{N}+\frac{2K}{N}\left(\frac{N^2\ln N}{2}-\frac{N^2}{4}\right)$$

$$= \alpha N+\frac{4\beta}{N}+KN\ln N-\frac{KN}{2}$$

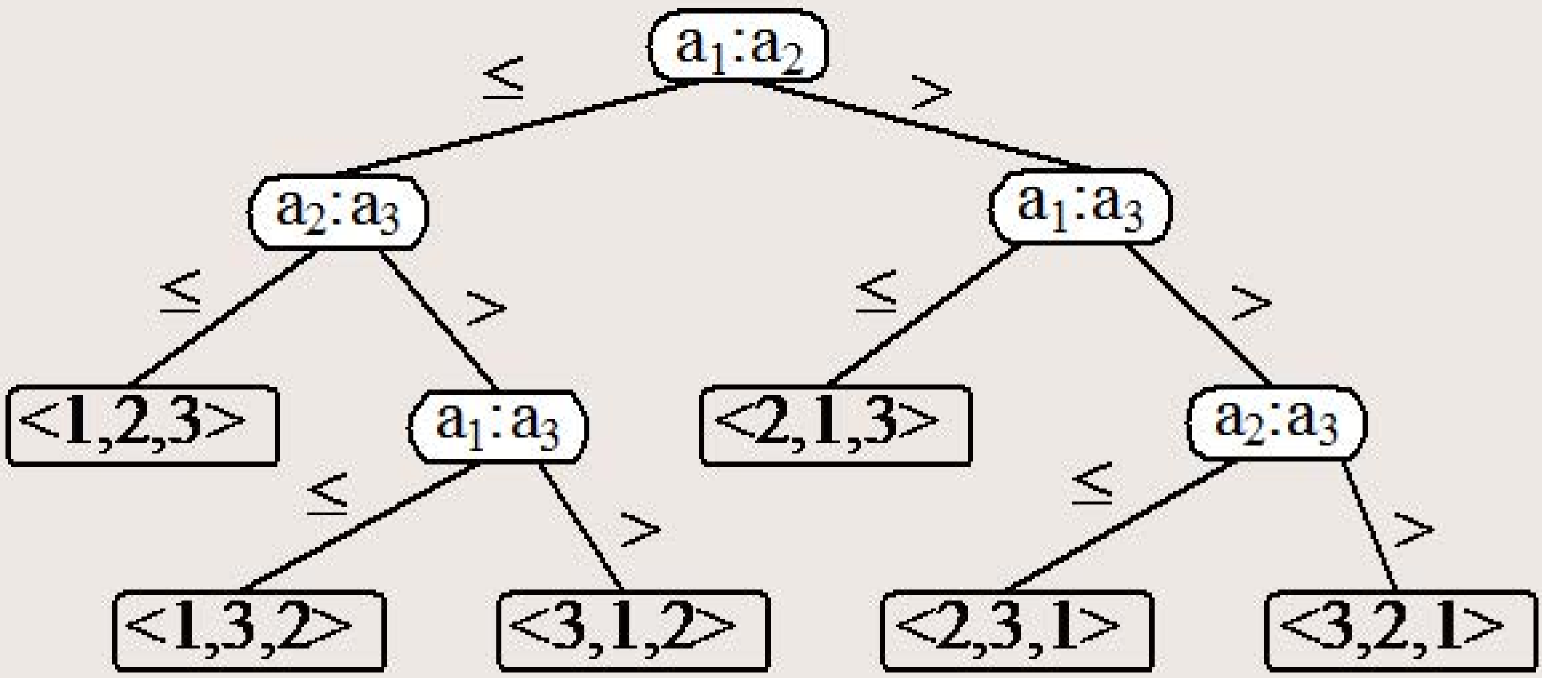
$$T\,к.\ \alpha N+\frac{4\beta}{N}\le\frac{KN}{2}\ (N\le 2)$$

$$\Rightarrow T(N)\le KN\ln N\le KN\log_2 N$$

Оценка сложности $T_{ср}= 1.4(N+1)\log_2 N$

Оценка минимальной сложности

Представим действия, выполняемые при сортировке данных, в виде <u>разрешающего дерева</u> (*decision tree*)



Теорема 3.1. Высота любого разрешающего дерева, сортирующего N элементов, есть $\Omega(N\log_2 N)$

Доказательство.

- Количество листьев N! (количество перестановок из N элементов)

- Двоичное дерево высоты h имеет 2^h листьев

$$\Rightarrow N! \le 2^h$$

- Используя формулу Стирлинга $N! > (N/e)^N$, можно заключить

$$h \ge N \log_2 N - N \log_2 e = \Omega(N\log_2 N)$$