УВАЖАЕМЫЕ СТУДЕНТЫ!

ВАМ НЕОБХОДИМО ВЫПОЛНИТЬ СЛЕДУЮЩЕЕ:

1. Ознакомиться с теорией и законспектировать лекцию не меньше трех листов, составить и ответить на вопросы.

2. Предоставит отчет конспекта лекции прислать в виде скриншото в течении трех дней .

3. Отправить преподавателю на почту **v.vika2014@mail.ru** и указать свою Ф.И.О, группу, и название дисциплины тел 072-17-44-9-22

Тема: Алгоритмы сортировки

Сортировка элементов в массиве

Сортировка представляет собой процесс упорядочения элементов в массиве в порядке возрастания или убывания их значений. Например, массив **X** из **n** элементов будет отсортирован в порядке возрастания значений его элементов, если $X_1 \leq X_2 \leq ... \leq X_n$, и в порядке убывания, если $X_1 \geq X_2 \geq ... \geq X_n$.

Существует большое количество алгоритмов сортировки, но все они базируются на трех основных:

- сортировка обменом;
- сортировка выбором;
- сортировка вставкой.

Представим, что нам необходимо разложить по порядку карты в колоде. Для сортировки карт **обменом** можно разложить карты на столе лицевой стороной вверх и менять местами те карты, которые расположены в неправильном порядке, делая это до тех пор, пока колода карт не станет упорядоченной.

Для сортировки **выбором** из разложенных на столе карт выбирают самую младшую (старшую) карту и держат ее в руках. Затем из оставшихся карт вновь выбрать наименьшую (наибольшую) по значению карту и помещают ее позади той карты, которая была выбрана первой. Этот процесс

повторяется до тех пор, пока вся колода не окажется в руках. Поскольку каждый раз выбирается наименьшая (наибольшая) по значению карта из оставшихся на столе карт, по завершению такого процесса карты будут отсортированы по возрастанию (убыванию).

Для сортировки **вставкой** из колоды берут две карты и располагают их в необходимом порядке по отношению друг к другу. Каждая следующая карта, взятая из колоды, должна быть установлена на соответствующее место по отношению к уже упорядоченным картам.

Итак, решим следующую задачу. Задан массив **Y** из **n** целых чисел. Расположить элементы массива в порядке возрастания их значений.

Сортировка методом "пузырька"

Сортировка пузырьковым методом является наиболее известной. Ее популярность объясняется запоминающимся названием, которое происходит из-за подобия процессу движения пузырьков в резервуаре с водой, когда каждый пузырек находит свой собственный уровень, и простотой алгоритма.

Сортировка методом "пузырька" использует метод обменной сортировки и основана на выполнении в цикле операций сравнения и при необходимости обмена соседних элементов. Рассмотрим алгоритм пузырьковой сортировки более подробно.

Сравним первый элемент массива со вторым, если первый окажется больше второго, то поменяем их местами. Те же действия выполним для второго и третьего, третьего и четвертого, **i**-го и **(i+1)**-го, **(n-1)**-го и n-го элементов. В результате этих действий самый большой элемент станет на последнее **n**-е место. Теперь повторим данный алгоритм сначала, но последний **n**-й элемент, рассматривать не будем, так как он уже занял свое место. После проведения данной операции самый большой элемент оставшегося массива станет на **(n-1)**-е место. Так повторяем до тех пор, пока не упорядочим весь массив.

В табл. 2 подробно расписан процесс упорядочивания элементов в массиве. Нетрудно заметить, что для преобразования массива, состоящего из

n элементов, необходимо просмотреть его **n-1** раз, каждый раз уменьшая диапазон просмотра на один элемент. Блок-схема описанного алгоритма приведена на рис. 8. Обратите внимание на то, что для перестановки элементов (блок 4) используется буферная переменная **b**, в которой временно хранится значение элемента, подлежащего замене

Таблица 2. Процесс упорядочивания элементов в массиве по возрастанию

Номер элемента	**1**	**2**	**3**	**4**	**5**
Исходный массив	7	3	5	4	2
Первый просмотр	3	5	4	2	7
Второй просмотр	3	4	2	5	7
Третий просмотр	3	2	4	5	7
Четвертый просмотр	2	3	4	5	7

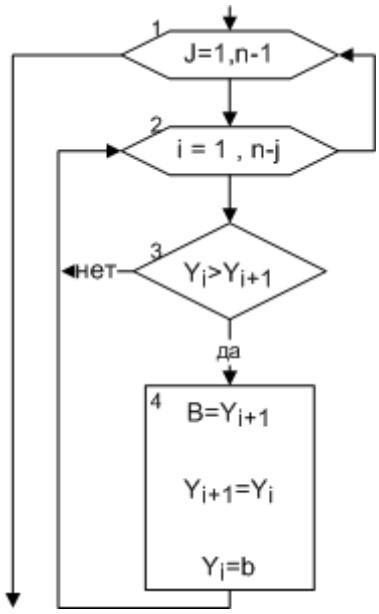

Рис. 8. Сортировка массива пузырьковым методом

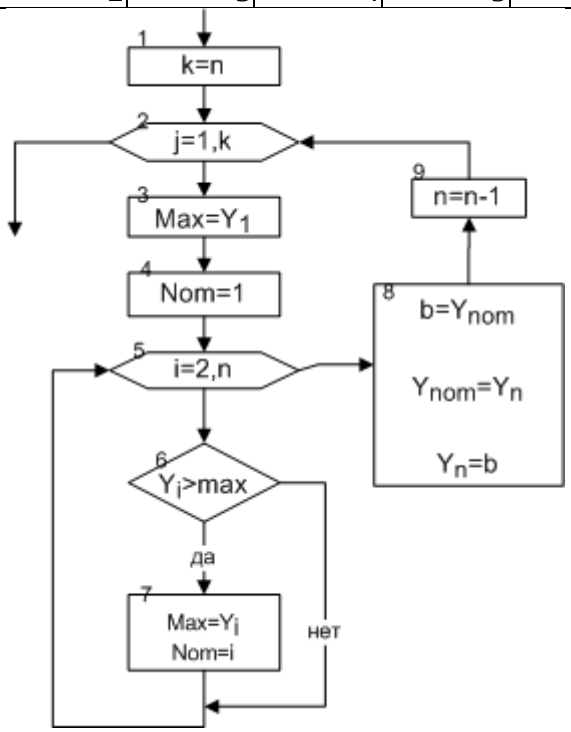

Рис. 9. Сортировка массива выбором наибольшего элемента

Сортировка выбором

Алгоритм **сортировки выбором** приведен в виде блок-схемы на рис. 9. Найдем в массиве самый большой элемент (блоки 3-7) и поменяем его местами с последним элементом (блок 8). Повторим алгоритм поиска максимального элемента, уменьшив количество просматриваемых элементов на единицу (блок 9), и поменяем его местами с предпоследним элементом

(блоки 3-7). Описанную выше операцию поиска проводим до полного упорядочивания элементов в массиве. Так как в блоке 9 происходит изменение переменной n, то в начале алгоритма ее значение необходимо сохранить (блок 1).

Сортировка вставкой

Сортировка вставкой заключается в том, что сначала упорядочиваются два элемента массива. Затем делается вставка третьего элемента в соответствующее место по отношению к первым двум элементам. Четвертый элемент помещают в список из уже упорядоченных трех элементов. Этот процесс повторяется до тех пор, пока все элементы не будут упорядочены.

Прежде чем приступить к составлению блок-схемы рассмотрим следующий пример. Пусть известно, что в массиве из восьми элементов первые шесть уже упорядочены, а седьмой элемент нужно вставить между вторым и четвертым. Сохраним седьмой элемент во вспомогательной переменной, так как показано на рисунке 10, а на его место запишем шестой. Далее пятый переместим на место шестого, четвертый на место пятого, а третий на место четвертого, тем самым, выполнив сдвиг элементов массива на одну позицию вправо. Записав содержимое вспомогательной переменной в третью позицию, достигнем нужного результата.

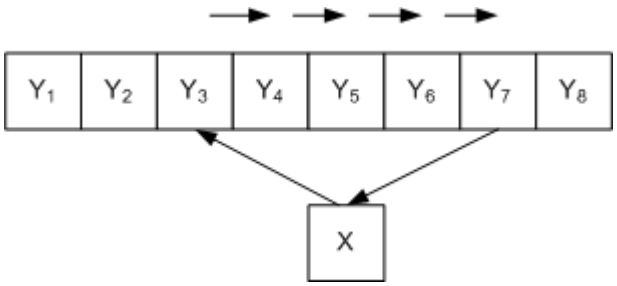

Рис. 10. Процесс вставки элемента в массив

Составим блок-схему алгоритма (рис.11), учитывая, что возможно описанные выше действия придется выполнить неоднократно.

Организуем цикл для просмотра всех элементов массива, начиная со второго (блок 4). Сохраним значение текущего **i**-го элемента во

вспомогательной переменной **X**, так как оно может быть потеряно при сдвиге элементов (блок 5) и присвоим переменной **j** значение индекса предыдущего **(i-1)**-го элемента массива (блок 6). Далее движемся по массиву влево в поисках элемента меньшего, чем текущий и пока он не найден сдвигаем элементы вправо на одну позицию. Для этого организуем цикл (блок 7), который прекратиться, как только будет найден элемент меньше текущего. Если такого элемента в массиве не найдется и переменная **j** станет равной нулю, то это будет означать, что достигнута левая граница массива, и текущий элемент необходимо установить в первую позицию. Смещение элементов массива вправо на одну позицию выполняется в блоке 8, а изменение счетчика **j** в блоке 9. Блок 10 выполняет вставку текущего элемента в соответствующую позицию.

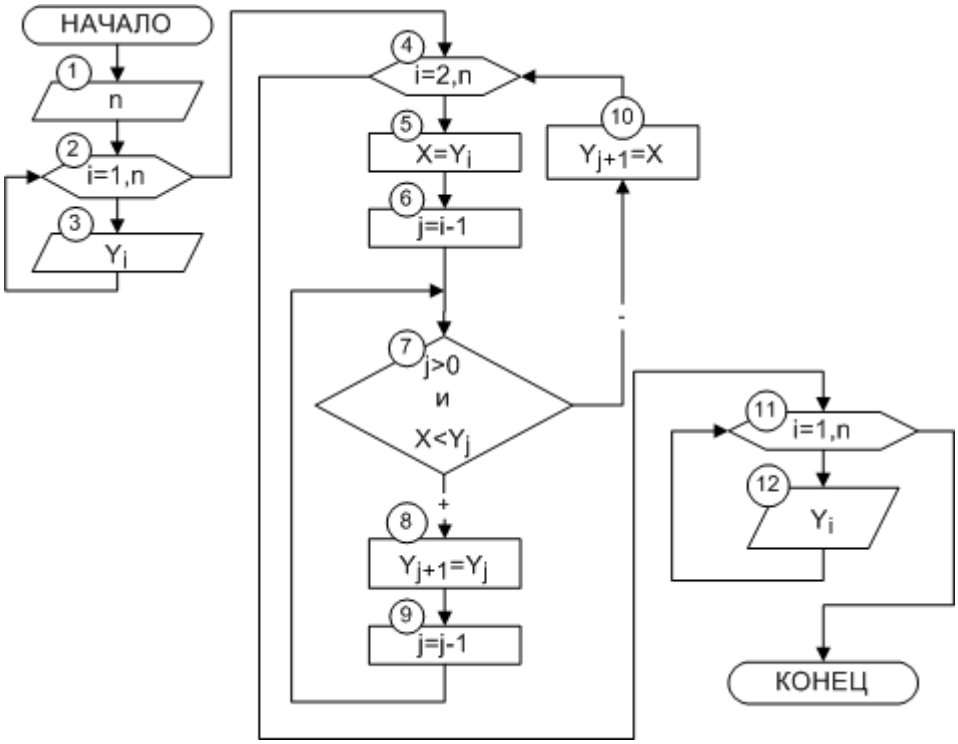

Рис. 11. Сортировка массива вставкой