

ЛАБОРАТОРНАЯ РАБОТА № 15

СОРТИРОВКА И ПОИСК В МАССИВАХ

Цель работы: приобретение практических навыков упорядочения (сортировки) массивов и поиска заданных элементов в массиве.

Краткие теоретические сведения

При решении задач, связанных с обработкой массивов, часто возникает необходимость поиска в массиве заданного элемента.

Задача. Определить, имеется ли в массиве, содержащем n чисел, элемент, значение которого совпадает с заданным числом L . Если такой элемент существует, надо указать его порядковый номер, в противном случае напечатать сообщение об отсутствии в массиве элемента, равного L .

Решение:

Для того, чтобы найти в массиве элемент, значение которого равно заданному числу L , необходимо просматривать подряд все элементы массива, начиная с первого, и сравнивать их значения с L . Просмотр массива надо продолжать до тех пор, пока не будет найден элемент массива, равный L , или до тех пор, пока не будет просмотрен весь массив (если такого элемента не существует).

Поиск нужного элемента может быть ускорен, если элементы массива упорядочены по своей величине, например в порядке возрастания значений или в алфавитном порядке.

Сортировка – это расстановка элементов некоторого списка в заданном порядке.

Существуют разные виды сортировки (по алфавиту, по датам и т.д.), они отличаются лишь процедурой сравнения элементов. Мы рассмотрим простейший вариант сортировки – расстановку элементов массива в порядке возрастания. Программи-

сты придумали множество методов сортировки. Они делятся на две группы:

- понятные, но не эффективные
- эффективные, но непонятные (быстрая сортировка и т.п.).

Пока мы будем изучать только методы из первой группы, которых хватает для простых задач (когда размер массива не более 1000).

Метод пузырька

Название этого метода произошло от известного физического явления – пузырек воздуха в воде поднимается вверх. В этом методе сначала поднимается «наверх» (к началу массива) самый «легкий» элемент (минимальный), затем следующий и т.д.

Сначала сравниваем последний элемент с предпоследним элементом. Если они стоят неправильно, то меняем их местами. Далее так же рассматриваем следующую пару элементов и т.д. Когда мы обработали пару ($A[0]$, $A[1]$), минимальный элемент стоит на месте $A[0]$. Это значит, что на следующих этапах его можно не рассматривать.

При следующем проходе наша задача – поставить на место элемент $A[1]$. Делаем это так же, но уже не рассматриваем $A[0]$, который стоит на своем месте. Сделав $N-1$ проходов, мы установим на место элементы с $A[0]$ по $A[N-2]$. Это значит, что последний элемент, $A[N-1]$, уже тоже стоит на своем месте (другого у него нет).

Метод пузырька работает медленно, особенно на больших массивах.

Метод выбора минимального элемента

Еще один недостаток метода пузырька состоит в том, что приходится слишком часто переставлять местами соседние элементы. Этого можно избежать, если использовать *метод выбора минимального элемента*. Он заключается в следующем. Ищем в массиве минимальный элемент и ставим его на первое место. Затем из оставшихся элементов также ищем минимальный и ставим на следующее место и т.д. В сравнении с методом пузырька, этот метод требует значительно меньше перестановок элементов (в

худшем случае N-1). Он дает значительный выигрыш, если перестановки сложны и занимают много времени.

Порядок выполнения работы

1. Изучить теоретические сведения.
2. Ответить на контрольные вопросы.
3. Выполнить задание.

Контрольные вопросы

1. Что означает термин "упорядочение массива"?
2. В чем заключается идея упорядочения массива с помощью поиска наименьшего элемента?
3. В чем заключается идея упорядочения массива методом пузырька?

Задания для выполнения

1. Заполнить массив из 10 элементов случайными числами в интервале [0..100] и отсортировать его по последней цифре.

Пример:

Исходный массив:

14 25 13 30 76 58 32 11 41 97

Результат:

30 11 41 32 13 14 25 76 97 58

2. Заполнить массив из 10 элементов случайными числами в интервале [0..100] и отсортировать первую половину по возрастанию, а вторую – по убыванию.

Пример:

Исходный массив:

14 25 13 30 76 58 32 11 41 97

Результат:

13 14 25 30 76 97 58 41 32 11