

Лабораторная работа №7

Сортировка одномерных массивов

Оглавление

Цель	1
Инструкция:.....	1
Задания для самостоятельного выполнения:.....	3
Домашнее задание:	4
БЛОК А	4
БЛОК В	4
БЛОК С	5
Требования к оформлению программ:.....	6
Контрольные вопросы:	6

Цель

Научиться сортировать одномерные массивы.

Инструкция:

Задача: упорядочить элементы одномерного массива по возрастанию. Использовать сортировку методом Пузырька.

Алгоритм сортировки:

1. Рассматриваем все еще неупорядоченные элементы массива.
2. Берем последний элемент массива и сравниваем его с предпоследним.
3. Меньший из них ставим на предпоследнее место и сравниваем его с элементом перед ним.
4. Выполняем до тех пор, пока минимальный не окажется на первом месте в массиве.
5. Неупорядоченные элементы в массиве начинаются со второго и т.д.
6. Повторяем, пока неупорядоченных элементов не останется

Решение: Использовать программу из Лабораторной работы 3, внести изменения в программу:

```

//-----main.cpp-----
//Лабораторная работа №7 Сортировка массивов

#include <iostream>

#include "vect.h"
#include "sort.h"

using namespace std;

int main()
{
    int n;
    double *v;

    do
    {
        cout<<"Enter massive size n>0(int), n = ";
        cin>>n;
    } while(n<=0);

    v= new double[n];

    InputDescVect(v,n);

    cout<<"Vector v:\n\t";
    OutputDescVect(v,n);

    BubbleSort(v,n);
    //SelectSort(v,n);
    //InsertSort(v,n);

    cout<<"Sorted vector v:\n\t";
    OutputDescVect(v,n);

    delete [] v; v=NULL;
}

```

```
//-----sort.h-----
```

```

#ifndef SORT_H_INCLUDED
#define SORT_H_INCLUDED

void BubbleSort (double *A, int N);
void SelectSort (double *A, int N);
void InsertSort (double *A, int N);

#endif // SORT_H_INCLUDED

```

```

//-----sort.cpp-----

void BubbleSort (double *A, int N)
{
    int i,j;
    double c;

    for ( i = 0; i < N-1; i ++ )
        for ( j = N-2; j >= i; j -- )
            if ( A[j] > A[j+1] )
            {
                c = A[j];   A[j] = A[j+1];  A[j+1] = c;
            }
}

void SelectSort (double *A, int N)
{
    int i, j, i_min;
    double c;

    for ( i = 0; i < N; i ++ )
    {
        i_min = i;
        for ( j = i+1; j < N-1; j ++ )
            if ( A[j] < A[i_min] )
                i_min = j;
        if ( i_min != i )
        {
            c = A[i];  A[i] = A[i_min];  A[i_min] = c;
        }
    }
}

void InsertSort (double *A, int N)
{
    int i, j;
    double x;

    for ( i = 1; i < N; i ++ )
    {
        x = A[i];
        j = i-1;

        while( x<A[j]  &&  j>=0  )
        {
            A[j+1] = A[j];
            j = j-1;
        }
        A[j+1] = x;
    }
}

```

Задания для самостоятельного выполнения:

Написать функции и продемонстрировать их работу:

1. Упорядочить элементы одномерного массива по возрастанию. Использовать сортировку *методом Пузырька*.

2. Упорядочить элементы одномерного массива по убыванию. Использовать сортировку **методом Пузырька**. Взять массив из n чисел, заполненный случайными числами из интервала [-10,10].

Домашнее задание:

БЛОК А

Задание на отметку «удовлетворительно»

Выполнить задание и оформить код программы в соответствии с требованиями. Подготовить письменные ответы на контрольные вопросы. В тетрадь выписывать вопрос и ответ на него.

Задание:

1. Упорядочить элементы одномерного массива по возрастанию. Использовать сортировку **методом выбора**.

Взять массив из n чисел, заполненный случайными числами из интервала [-10,10].

2. Упорядочить элементы одномерного массива по возрастанию. Использовать сортировку **методом вставки**.

Взять массив из n чисел, заполненный случайными числами из интервала [-10,10].

БЛОК В

Задания на отметку «хорошо»

Выполнить задания:

1. Упорядочить элементы одномерного массива по возрастанию. Использовать сортировку **методом Пузырька**. Посчитать количество сравнений и перестановок.

Количество сравнений – сколько раз сравнивались элементы массива

Количество перестановок - сколько раз элементы массив переставлялись

Выполнить задание:

- a. для массива из n чисел, заполненного по правилу: $n \ (n-1) \ (n-2) \ \dots \ 3 \ 2 \ 1$, где $n=10$, $n=100$, $n=1000$. Сравнить полученное количество сравнений и перестановок с теоретически полученными оценками:

Количество сравнений в «худшем случае»: $(n^2-n)/2$

Количество перестановок в «худшем случае»: $3/2(n^2-n)$

- b. Заполнить таблицу:

Размер массива	Количество сравнений		Количество перестановок	
	полученное	теоретическая оценка	полученное	теоретическая оценка
$n=10$				
$n=100$				
$n=1000$				

- c. Выполнить задание **1.a.** для массива из **n** чисел, заполненного случайными числами из интервала [-10,10]. Заполнить сравнительную таблицу.
2. Упорядочить элементы одномерного массива по возрастанию. Использовать сортировку **методом выбора**. Посчитать количество сравнений и перестановок. Взять массив из **n** чисел, заполненный по правилу: n (n-1) (n-2) ... 3 2 1, где n=10, n=100, n=1000.

Сравнить полученное количество сравнений и перестановок с теоретически полученными оценками:

$$\text{Количество сравнений в «худшем случае»: } (n^2-n)/2$$

$$\text{Количество перестановок в «худшем случае»: } n^2/4-3(n-1)$$

Заполнить сравнительную таблицу.

3. Упорядочить элементы одномерного массива по возрастанию. Использовать сортировку **методом вставки**. Посчитать количество сравнений и перестановок. Взять массив из **n** чисел, заполненный по правилу: n (n-1) (n-2) ... 3 2 1, где n=10, n=100, n=1000.

Сравнить полученное количество сравнений и перестановок с теоретически полученными оценками:

$$\text{Количество сравнений в «худшем случае»: } 1/4(n^2+n-4)$$

$$\text{Количество перестановок в «худшем случае»: } 1/2 (n^2+3n - 4).$$

Заполнить сравнительную таблицу.

БЛОК С

Задания на отметку «отлично»

Выполнить задание из **блока В** и рассмотреть еще один любой метод сортировки из предложенного списка: Челночная сортировка, метод Шелла, сортировка Хоара.

Упорядочить элементы одномерного массива по возрастанию. Использовать выбранный метод сортировки. Посчитать количество сравнений и перестановок.

Количество сравнений – сколько раз сравнивались элементы массива

Количество перестановок - сколько раз элементы массива переставлялись

Выполнить задание:

- a. для массива из **n** чисел, заполненного по правилу: n (n-1) (n-2) ... 3 2 1, где n=10, n=100, n=1000.
 b. Заполнить таблицу:

Размер массива	Количество сравнений		Количество перестановок	
	полученное	теоретическая оценка	полученное	теоретическая оценка
n=10				

n=100				
n=1000				

- c. Выполнить задание **1.a.** для массива из ***n*** чисел, заполненного случайными числами из интервала [-10,10]. Заполнить сравнительную таблицу.

Требования к оформлению программ:

1. **Содержание.** Программа должна делать то, что предусмотрено заданием. Не надо выполнять лишних действий, заданием не предусмотренных.
 2. **Спецификация.** В преамбуле программы в комментариях указывать сведения:
 - Кто выполнил.
 - Что делает программа (кратко).
 - Что на входе (имена входных файлов и т.д.).
 - Что на выходе (что является результатом работы программы?).
 3. **Ввод и вывод**
 - Приглашения к вводу (например, сколько чисел, какого типа и через какой разделитель нужно вводить).
 - Контрольный вывод (все введенные данные выводить на экран, и только после этого выполнять необходимые вычисления.)
 - «Защита от дурака». Проверять вводимые данные на корректность. Например, если необходимо считать количество чего – то, то эта величина не может быть отрицательной и т.д.
 4. **Структура кода.** Набираемый код должен быть хорошо структурированным.
Использовать:
 - Отступы.
 - Комментарии – поясняют решение программы.
 - Осмысленные названия переменных.
 - Пояснения о назначении переменных в комментариях (кроме счетчиков).
 5. **Декомпозиция кода**
 - Функциональная. Программу оформлять с помощью функций.
 6. **Многофайловые проекты**
 - Все проекты должны состоять минимум из двух модулей: главного и подключаемого.
- Главный модуль `main.cpp`.** В нем оставить только функцию `main`, в которой вызывать функции, описанные в других модулях.
- Модуль с описанием пользовательских функций `file.cpp` и `file.h`.** В этот модуль перенести определение всех функций, необходимых для выполнения задания.

Контрольные вопросы:

1. Что такое сортировка массива?
2. Напишите алгоритм сортировки последовательности чисел методом Пузырька по убыванию
3. Напишите функцию для сортировки массива вещественных чисел (float) методом Пузырька по убыванию. Нарисуйте блок-схему для данной функции.

4. Напишите алгоритм сортировки последовательности чисел методом выбора по убыванию
5. Напишите функцию для сортировки массива вещественных чисел (float) методом выбора по убыванию. Нарисуйте блок-схему для данной функции.
6. Напишите алгоритм сортировки последовательности чисел методом вставки по убыванию
7. Напишите функцию для сортировки массива вещественных чисел (float) методом вставки по убыванию. Нарисуйте блок-схему для данной функции.
8. Какие «критерии эффективности» метода сортировки можно обычно проверяют?
9. Перечислите достоинства и недостатки использования метода Пузырька для сортировки массивов.
10. Перечислите достоинства и недостатки использования метода выбора для сортировки массивов.
11. Перечислите достоинства и недостатки использования метода вставки для сортировки массивов.
12. Приведите и другие методы сортировки, кроме метода Пузырька, вставки и выбора. (не менее четырех) .