

|  |  |
| --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное бюджетное образовательное учреждение высшего образования  **«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ГОСУДАРСТВЕННЫЙ**  **ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»** |  |

**Кафедра «Высшая математика»**

**Лабораторная работа № 3**

по дисциплине

«Структуры и алгоритмы обработки данных»

на тему:

**«**Сортировки и определение характеристик одномерных массивов**»**

**Выполнил:**

Учебная группа: 1бПМ   
ФИО: Кузнецов А.-С. О.

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Руководитель лабораторной работы:**

Должность: старший преподаватель

Звание: б/з

ФИО: Кутейников И. А.

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Цель: определить массив целых чисел (например, размера 30),

заполнить его случайными числами (в диапазоне от A до B) или ввести его

элементы с клавиатуры и определить его характеристики в соответствии с

Программа должна содержать следующие функции:

1 Инициализация элементов массива числами, вводимыми с

клавиатуры;

2 Инициализация элементов массива случайными числами;

3 Вывод массива на экран;

4 Сортировка массива;

4.1 Сортировка в соответствии с вариантом;

4.2 Собственная реализация быстрой сортировки;

4.3 Сортировка встроенной функцией сортировки;

4.4 Сравнение времени работы сортировок;

5 Решение задачи уровня Б;

6 Решение задачи уровня С (опционально);

7 Выход из программы.

Реализовать заданный метод сортировки числового массива в

соответствии с индивидуальным заданием. Для всех вариантов добавить

реализацию быстрой сортировки (quicksort). Оценить время работы каждого

алгоритма сортировки и сравнить его со временем стандартной функции

сортировки, используемой в выбранном языке программирования.

Уровень В

B5. Определить порядковые номера трех наименьших элементов и их сумму.

Уровень С

Определить массив целых чисел a (например, размера 30),

заполнить его случайными числами (в диапазоне от A до B) или ввести его

элементы с клавиатуры.

Определить количество попарно различных элементов массива a,

обладающих свойством Q (рассмотреть три случая: произвольный массив,

упорядоченный массив, массив с достаточно узким диапазоном значений

элементов). Свойства Q задаются в вариантах.

Программа должна содержать следующие функции:

• инициализация элементов массива случайными числами или вводимыми с

клавиатуры;

• вывод массива на экран;

• количество попарно различных элементов со свойством Q для

произвольного массива;

• количество попарно различных элементов со свойством Q для

упорядоченного массива;

• количество попарно различных элементов со свойством Q для массива с

достаточно узким диапазоном значений элементов.

Варианты заданий

C5. Q: число является нечетным.

Код: (Здесь приведен не весь код, а лишь важная его часть)

static int[] fillByHand ()  
{  
 int size =0;  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.print("Введите размер массива: ");  
 size=in.nextInt();  
 int[] mass= new int[size];  
 for (int i = 0; i<mass.length;i++)  
 {  
 System.*out*.println("Введите "+i+"-й элемент: ");  
 mass[i]=in.nextInt();  
 }  
 return mass;  
}  
static int[] fillByRandom()  
{  
 int size=0;  
 int leftB,rightB;  
 Scanner in= new Scanner(System.*in*);  
 System.*out*.println("Введите размер массива: ");  
 size= in.nextInt();  
 int [] mass = new int [size];  
 System.*out*.println("Введите левую границу: ");  
 leftB= in.nextInt();  
 System.*out*.println("Введите правую границу: ");  
 rightB=in.nextInt();  
 Random rnd= new Random();  
 for (int i = 0; i < mass.length; i ++)  
 {  
 mass[i]= rnd.nextInt(rightB-leftB)+leftB;  
 }  
 return mass;  
}  
static int [] BubbleSort(int [] mass){  
 int buf = 0;  
 boolean swap;  
 int counter = 0;  
 for (int i = 0; i < mass.length; i++) {  
 counter=i+1;  
 swap= false;  
 for (int j = 1;j < (mass.length-i); j++) {  
 if (mass[j-1]>mass[j])  
 {  
 buf=mass[j];  
 mass[j]=mass[j-1];  
 mass[j-1]=buf;  
 swap= true;  
 }  
 }  
 if (swap==false) {  
 break;}  
 }  
 System.*out*.println("BubleSort - complete");  
 return mass;  
}  
static void QuickSort(int [] mass, int left , int right){  
 if (mass.length==0) {  
 return;  
 }  
 if (left>=right) {  
 return;  
 }  
 int mid= left +(right- left)/2;  
 int op = mass[mid];  
 int i= left, j=right;  
 while(i<=j)  
 {  
 while (mass[i]<op){  
 i++;  
 }  
 while(mass[j]>op){  
 j--;  
 }  
 if (i<=j){  
 int buf= mass[i];  
 mass[i]=mass[j];  
 mass[j]=buf;  
 i++;  
 j--;  
 }  
 if (left<j){  
 *QuickSort*(mass,left,j);  
 }  
 if (right>i){  
 *QuickSort*(mass,i,right);  
 }  
 }  
}

Результат:

1.Задать массив.

2.Задать рандомный массив.

3.Вывести массив в консоль.

4.Сортировать массив.

5.Решение задачи уровня B.

6.Решение задачи уровня С.

7.Выйти из меню.

Выберите пункт меню: 2

Введите размер массива:

5

Введите левую границу:

1

Введите правую границу:

10

Массив:

2 9 5 5 6

1.Задать массив.

2.Задать рандомный массив.

3.Вывести массив в консоль.

4.Сортировать массив.

5.Решение задачи уровня B.

6.Решение задачи уровня С.

7.Выйти из меню.

Выберите пункт меню: 4

4.1. Сортировка в соответствии с вариантом 5.

4.2. Быстрая сортировка.

4.3. Сортировка встроенной функцией сортировки.

4.4. Сравнение времени работы сортировок.

выберите пункт меню: 1

BubleSort - complete

SortTime=0

2 5 5 6 9

Quick-complete

SortTime=0

2 5 5 6 9

Funcinjava-complete

SortTime= 0

2 5 5 6 9

1.Задать массив.

2.Задать рандомный массив.

3.Вывести массив в консоль.

4.Сортировать массив.

5.Решение задачи уровня B.

6.Решение задачи уровня С.

7.Выйти из меню.

Выберите пункт меню: 5

sum= 12

index1=0

index2=1

index3=1

1.Задать массив.

2.Задать рандомный массив.

3.Вывести массив в консоль.

4.Сортировать массив.

5.Решение задачи уровня B.

6.Решение задачи уровня С.

7.Выйти из меню.

Выберите пункт меню: 6

count: 0

Вывод

В результате выполнения лабораторной работы появилось понимание того, как работать с массивами, а именно:

1. Заполнять его вручную
2. Заполнять его рандомными значениями.
3. Сортировать его оптимальным методом и смотреть время работы сортировки
4. Решать определенные задачи с использованием массива
5. Находить определенную характеристическую функцию массива

Также было получено лучшее понимание синтаксиса языка.