Министерство науки и высшего образования Российской Федерации

Федеральное государственное образовательное учреждение

высшего образования

**ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

**Дисциплина: Алгоритмы и структуры данных**

**Отчёт по лабораторной работе №8**

**Тема: «Рекурсия. Алгоритм быстрой сортировки. Алгоритм сортировки слиянием»**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил студент  группы ИТб 2302-02-20 |  | Ердяков Роман Александрович |  |
|  |  |
| Проверила |  | Кашина Елена Вячеславовна |  |

СОДЕРЖАНИЕ

[1 Задание 3](#_Toc200883346)

[2 Тестирование (задача 1) 4](#_Toc200883347)

[3 Программа на C# (задача 1): 5](#_Toc200883348)

[4 Тестирование (задача 2) 6](#_Toc200883349)

[5 Программа на C# (задача 2): 7](#_Toc200883350)

[6 Тестирование (задача 3) 8](#_Toc200883351)

[7 Программа на C# (задача 3): 9](#_Toc200883352)

[8 Вывод 11](#_Toc200883353)

# Задание

Задача 1.

Дан одномерный массив. Отсортировать его методом Хоара по неубыванию, используя рекурсию.

Задача 2.

Сформировать из двух одномерных массивов третий, используя метод слияния.

Задача 3.

Отсортировать одномерный массив методом слияния, используя рекурсию.

# Тестирование (задача 1)

Примеры выполнения программы представлены на рисунке 1 и 2.

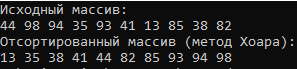


Рисунок 1 – Экранная форма программы

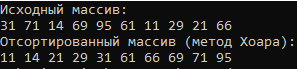


Рисунок 2 – Экранная форма программы

# Программа на C# (задача 1):

class Program

{

static void Main()

{

Random rand = new Random();

int[] array = new int[10];

Console.WriteLine("Исходный массив:");

for (int i = 0; i < array.Length; i++)

{

array[i] = rand.Next(0, 100);

Console.Write(array[i] + " ");

}

QuickSort(array, 0, array.Length - 1);

Console.WriteLine("\nОтсортированный массив (метод Хоара):");

foreach (int item in array)

{

Console.Write(item + " ");

}

}

static void QuickSort(int[] arr, int left, int right)

{

if (left >= right) return;

int pivot = arr[(left + right) / 2];

int index = Partition(arr, left, right, pivot);

QuickSort(arr, left, index - 1);

QuickSort(arr, index, right);

}

static int Partition(int[] arr, int left, int right, int pivot)

{

while (left <= right)

{

while (arr[left] < pivot) left++;

while (arr[right] > pivot) right--;

if (left <= right)

{

(arr[left], arr[right]) = (arr[right], arr[left]);

left++;

right--;

}

}

return left;

}

}

# Тестирование (задача 2)

Примеры выполнения программы представлены на рисунках 3, 4.

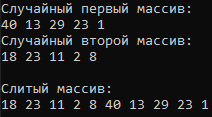


Рисунок 3 – Экранная форма программы

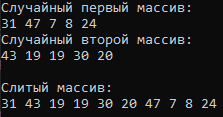


Рисунок 4 – Экранная форма программы

# Программа на C# (задача 2):

class Program

{

static void Main()

{

Random rand = new Random();

int[] arr1 = new int[5];

int[] arr2 = new int[5];

for (int i = 0; i < 5; i++)

{

arr1[i] = rand.Next(1, 50);

arr2[i] = rand.Next(1, 50);

}

Console.WriteLine("Случайный первый массив:");

PrintArray(arr1);

Console.WriteLine("Случайный второй массив:");

PrintArray(arr2);

int[] merged = new int[arr1.Length + arr2.Length];

MergeRecursive(arr1, 0, arr2, 0, merged, 0);

Console.WriteLine("\nСлитый массив:");

PrintArray(merged);

}

static void MergeRecursive(int[] arr1, int i, int[] arr2, int j, int[] merged, int k)

{

if (i == arr1.Length && j == arr2.Length)

return;

if (i == arr1.Length)

{

merged[k] = arr2[j];

MergeRecursive(arr1, i, arr2, j + 1, merged, k + 1);

}

else if (j == arr2.Length)

{

merged[k] = arr1[i];

MergeRecursive(arr1, i + 1, arr2, j, merged, k + 1);

}

else if (arr1[i] < arr2[j])

{

merged[k] = arr1[i];

MergeRecursive(arr1, i + 1, arr2, j, merged, k + 1);

}

else

{

merged[k] = arr2[j];

MergeRecursive(arr1, i, arr2, j + 1, merged, k + 1);

}

}

static void PrintArray(int[] array)

{

foreach (var item in array)

Console.Write(item + " ");

Console.WriteLine();

}

}

# Тестирование (задача 3)

Примеры выполнения программы представлены на рисунках 6, 7.

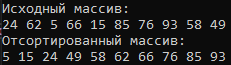


Рисунок 6 – Экранная форма программы

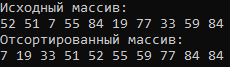


Рисунок 7 – Экранная форма программы

# Программа на C# (задача 3):

class Program

{

static void Main()

{

Random rand = new Random();

int[] array = new int[10];

for (int i = 0; i < array.Length; i++)

array[i] = rand.Next(1, 100);

Console.WriteLine("Исходный массив:");

PrintArray(array);

int[] sorted = MergeSort(array);

Console.WriteLine("Отсортированный массив:");

PrintArray(sorted);

}

static int[] MergeSort(int[] array)

{

if (array.Length <= 1)

return array;

int mid = array.Length / 2;

int[] left = new int[mid];

int[] right = new int[array.Length - mid];

Array.Copy(array, 0, left, 0, mid);

Array.Copy(array, mid, right, 0, array.Length - mid);

left = MergeSort(left);

right = MergeSort(right);

return Merge(left, right);

}

static int[] Merge(int[] left, int[] right)

{

int[] result = new int[left.Length + right.Length];

int i = 0, l = 0, r = 0;

while (l < left.Length && r < right.Length)

{

if (left[l] < right[r])

result[i++] = left[l++];

else

result[i++] = right[r++];

}

while (l < left.Length)

result[i++] = left[l++];

while (r < right.Length)

result[i++] = right[r++];

return result;

}

static void PrintArray(int[] array)

{

foreach (var item in array)

Console.Write(item + " ");

Console.WriteLine();

}

}

# Вывод

В ходе выполнения этих задач освоили основные методы работы с массивами и рекурсией, закрепили понимание рекурсивного подхода к сортировке и объединению данных, а также важность правильного деления массива и слияния для оптимальной работы алгоритмов.