Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра програмування та захисту інформації

Звіт

з виконаної лабораторної роботи № 3

дисципліни “ Технології розробки алгоритмів ”

на тему

“ Алгоритми сортування ”

Виконав

студент академічної

групи КІ-15

Аннаев А

Перевірив

Викл.

Гермак В.С.

Кропивницкий- 2017

Лабораторная работа №3

Тема: Алгоритмы сортировки

Цель: Розлянуты алгоритмы сортировки, исследовать целесообразность применения различных алгоритмов в конкретных случаях исходя из их эффективности

Ход работы.

Задание

Вариант 8 : Реализовать рекурсивную версию алгоритма Хоара.

Отличительной особенностью быстрой сортировки является операция разбиения массива на две части относительно опорного элемента. Например, если последовательность требуется упорядочить по возрастанию, то в левую часть будут помещены все элементы, значения которых меньше значения опорного элемента, а в правую элементы, чьи значения больше или равны опорному. Вне зависимости от того, какой элемент выбран в качестве опорного, массив будет отсортирован, но все же наиболее удачным считается ситуация, когда по обеим сторонам от опорного элемента оказывается примерно равное количество элементов. Если длина какой-то из получившихся в результате разбиения частей превышает один элемент, то для нее нужно рекурсивно выполнить упорядочивание, т. е. повторно запустить алгоритм на каждом из отрезков.

Таким образом, алгоритм быстрой сортировки включает в себя два основных этапа:

разбиение массива относительно опорного элемента;

рекурсивная сортировка каждой части массива.

static void Main(string[] args)

{

int min = 0;

int max = 100;

int[] array = new int[max]; // сортируемый массив

Random rand = new Random();

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

array[i] = rand.Next(min, max);

Console.WriteLine("Массив до сортировки");

DisplayArray(array);

Console.WriteLine("Массив после сортировки");

HoarAlgorithms(array, min, array.Length - 1);

DisplayArray(array);

Console.ReadKey();

}

/// <summary>

/// Рекурсивная версия алгоритма Хоара

/// </summary>

/// <param name="data">массив для сортировки</param>

/// <param name="begin"></param>

/// <param name="end"></param>

private static int partition(int[] array, int start, int end)

{

int marker = start;

for (int i = start; i <= end; i++)

{

if (array[i] <= array[end])

{

int temp = array[marker]; // swap

array[marker] = array[i];

array[i] = temp;

marker += 1;

}

}

return marker - 1;

}

/// <summary>

/// Алгоритм Хоара

/// </summary>

/// <param name="array">массив</param>

/// <param name="start">мин</param>

/// <param name="end">макс</param>

private static void HoarAlgorithms(int[] array, int start, int end)

{

if (start >= end)

{

return;

}

int pivot = partition(array, start, end);

HoarAlgorithms(array, start, pivot - 1);

HoarAlgorithms(array, pivot + 1, end);

}

/// <summary>

/// Отображения содержимого массива в консоли

/// </summary>

/// <param name="array">массив для отображения</param>

private static void DisplayArray(int [] array)

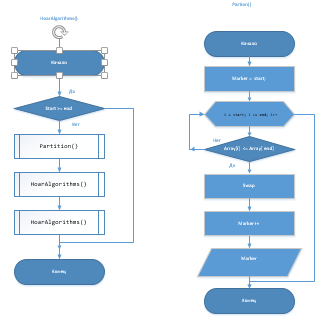
{

foreach (int value in array)

Console.WriteLine(Convert.ToString(value));

}

|  |  |  |  |
| --- | --- | --- | --- |
| Худшее время | O(*n*2) |  |  |
| Лучшее время | O(*n* log *n*) |  |  |
| Среднее время | O(*n* log *n*) |  |  |



Выводы:

В ходе выполнения лабораторной работы № 3 я рассмотрел алгоритмы сортировки, исследовал целесообразность применения различных алгоритмов в конкретных случаях исходя из их эффективности и реализовала сортировки Хоара.