Министерство науки и высшего образования

Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего образования

«Новосибирский государственный технический университет»



Кафедра теоретической и прикладной информатики

Лабораторная работа № 6  
по дисциплине «Метрология, качество и тестирование программного обеспечения»

**Вычисление метрических характеристик реализаций алгоритмов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Факультет: | ПМИ |  |  |
| Группа: | ПМИМ-01 |  |  |
| Студент: | Ничипиенко В.А. |  |  |
| Вариант: | 6 |  |  |
| Преподаватель: | Зайцев М.Г. |  |  |

Новосибирск

2021

1. **Цель работы**

Научиться вычислять метрические характеристики реализаций алгоритмов

1. **Задание**

Написать подпрограммы на двух языках программирования для решения

следующих задач

* Отыскать минимальный элемент одномерного массива целых, его значение и значение его индекса.
* Сортировка одномерного массива в порядке возрастания методом пузырька.
* Бинарный поиск элемента в упорядоченном одномерном массиве.
* Отыскать минимальный элемент двумерного массива целых, его значение и значение его индексов.
* Осуществить перестановку значений элементов одномерного массива в обратном порядке.
* Осуществлять циклический сдвиг элементов одномерного массива на заданное число позиций влево.
* Заменить все вхождения целочисленного значения в целочисленный массив.

1. **Исходные тексты программ**

*Реализация на C#*

*Текст программы Algorithms.cs*

using System;

namespace Algorithms

{

class Algorithms

{

public void FindMinInArray(int[] array)

{

int min = int.MaxValue;

int index = 0;

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

{

if (array[i] < min)

{

min = array[i];

index = i;

}

}

Console.WriteLine($"Min elem = {min}, index = {index}");

}

public int[] BubbleSort(int[] array)

{

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

{

for (int j = i + 1; j < array.Length; j++)

{

if (array[i] > array[j])

{

int temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

}

return array;

}

public int BinarySearch(int[] array, int elem)

{

return Array.BinarySearch(array, elem);

}

public void FindMinInMatrix(int[,] matrix)

{

int rows = matrix.GetUpperBound(0) + 1;

int columns = matrix.Length / rows;

int min = int.MaxValue;

int index = 0;

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

{

for (int j = 0; j < columns; j++)

{

if (matrix[i, j] < min)

{

min = matrix[i, j];

index = i;

}

}

}

Console.WriteLine($"Min elem = {min}, index = {index}");

}

public int[] ReverseArray(int[] array)

{

for (int i = 0, j = array.Length - 1; i < j; i++, j--)

{

var t = array[i];

array[i] = array[j];

array[j] = t;

}

return array;

}

public int[] ShiftArrayElements(int[] array, int shiftParameter)

{

int[] newArray = new int[array.Length];

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

{

newArray[i] = array[(i + shiftParameter) % newArray.Length];

}

return newArray;

}

public int[] ReplaceElements(int[] array, int elem, int value)

{

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

{

if (array[i] == elem)

{

array[i] = value;

}

}

return array;

}

}

}

*Реализация на C++*

*Текст программы Algorithms.h*

#pragma once

#include <tuple>

using namespace std;

class Algorithms

{

public:

tuple<int, int> FindMinInArray(int\* array, int size);

int\* BubbleSort(int\* array, int size);

int BinarySearch(int array[], int size, int elem);

tuple<int, int> FindMinInMatrix(int(\*)[4], int);

int\* ReverseArray(int\* array, int size);

int\* ShiftArrayElements(int\* array, int size, int shiftParameter);

int\* ReplaceElements(int\* array, int size, int elem, int value);

};

*Текст программы Algorithms.cpp*

#include "Algorithms.h"

#include <limits.h>

#include <iostream>

using namespace std;

tuple<int, int> Algorithms::FindMinInArray(int\* array, int size)

{

int min = INT\_MAX;

int index = 0;

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

{

if (array[i] < min)

{

min = array[i];

index = i;

}

}

return tuple<int, int>(min, index);

}

int\* Algorithms::BubbleSort(int\* array, int size)

{

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

{

for (int j = i + 1; j < size; j++)

{

if (array[i] > array[j])

{

int temp = array[i];

array[i] = array[j];

array[j] = temp;

}

}

}

return array;

}

int Algorithms::BinarySearch(int array[], int size, int elem)

{

int left = 0, right = size, mid = 0;

while (left <= right)

{

mid = left + (right - left) / 2;

if (elem < array[mid])

right = mid - 1;

else

if (elem > array[mid])

left = mid + 1;

else

return mid;

}

return -1;

}

tuple<int, int> Algorithms::FindMinInMatrix(int(\*matrix)[4], int rows)

{

int columns = sizeof(\*matrix) / sizeof(\*matrix[0]);

int min = INT\_MAX;

int index = 0;

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

{

for (int j = 0; j < columns; j++)

{

if (matrix[i][j] < min)

{

min = matrix[i][j];

index = i;

}

}

}

return tuple<int, int>(min, index);

}

int\* Algorithms::ReverseArray(int\* array, int size)

{

for (int i = 0, j = size - 1; i < j; i++, j--)

{

int t = array[i];

array[i] = array[j];

array[j] = t;

}

return array;

}

int\* Algorithms::ShiftArrayElements(int\* array, int size, int shiftParameter)

{

int \*newArray = new int[size];

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

{

newArray[i] = array[(i + shiftParameter) % size];

}

return newArray;

}

int\* Algorithms::ReplaceElements(int\* array, int size, int elem, int value)

{

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

{

if (array[i] == elem)

{

array[i] = value;

}

}

return array;

}

1. **Метрические характеристики реализации и алгоритма**
2. Отыскать минимальный элемент одномерного массива целых, его значение и значение его индекса.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n\*2 | n1 | n2 | n | N1 | N2 | N | N^ | V\* |
| C# | 3 | 5 | 4 | 9 | 9 | 8 | 17 | 19,609 | 11,609 |
| C++ | 4 | 5 | 4 | 9 | 9 | 8 | 17 | 19,609 | 15,509 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V | L | L^ | I | T^1 | T^2 | T^3 | λ1 | λ2 |
| C# | 53,888 | 0,215 | 0,2 | 10,777 | 13,896 | 17,266 | 14,969 | 2,501 | 2,501 |
| C++ | 53,888 | 0,287 | 0,2 | 10,777 | 10,402 | 17,266 | 14,969 | 4,463 | 4,463 |

1. Сортировка одномерного массива в порядке возрастания методом пузырька.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n\*2 | n1 | n2 | n | N1 | N2 | N | N^ | V\* |
| C# | 2 | 4 | 6 | 10 | 8 | 7 | 15 | 23,509 | 8 |
| C++ | 3 | 4 | 6 | 10 | 8 | 7 | 15 | 23,509 | 11,609 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V | L | L^ | I | T^1 | T^2 | T^3 | λ1 | λ2 |
| C# | 49,828 | 0,160 | 0,428 | 21,355 | 17,242 | 10,123 | 6,459 | 1,284 | 1,284 |
| C++ | 49,828 | 0,232 | 0,428 | 21,355 | 11,881 | 10,123 | 6,459 | 2,704 | 2,704 |

1. Бинарный поиск элемента в упорядоченном одномерном массиве.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n\*2 | n1 | n2 | n | N1 | N2 | N | N^ | V\* |
| C# | 3 | 2 | 2 | 4 | 2 | 2 | 4 | 4 | 11,609 |
| C++ | 4 | 5 | 5 | 10 | 12 | 7 | 19 | 23,219 | 15,509 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V | L | L^ | I | T^1 | T^2 | T^3 | λ1 | λ2 |
| C# | 8 | 1,451 | 1 | 8 | 0,306 | 0,444 | 0,444 | 16,847 | 16,847 |
| C++ | 63,116 | 0,245 | 0,285 | 18,033 | 14,269 | 14,998 | 12,272 | 3,811 | 3,811 |

1. Отыскать минимальный элемент двумерного массива целых, его значение и значение его индексов.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n\*2 | n1 | n2 | n | N1 | N2 | N | N^ | V\* |
| C# | 3 | 5 | 6 | 11 | 14 | 10 | 24 | 27,119 | 11,609 |
| C++ | 4 | 5 | 6 | 11 | 14 | 10 | 24 | 27,119 | 15,509 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V | L | L^ | I | T^1 | T^2 | T^3 | λ1 | λ2 |
| C# | 83,026 | 0,139 | 0,24 | 19,926 | 32,986 | 21,717 | 19,219 | 1,623 | 1,623 |
| C++ | 83,026 | 0,186 | 0,24 | 19,926 | 24,691 | 21,717 | 19,219 | 2,897 | 2,897 |

1. Осуществить перестановку значений элементов одномерного массива в обратном порядке.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n\*2 | n1 | n2 | n | N1 | N2 | N | N^ | V\* |
| C# | 2 | 4 | 5 | 9 | 6 | 5 | 11 | 19,609 | 8 |
| C++ | 3 | 4 | 5 | 9 | 6 | 5 | 11 | 19,609 | 11,609 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V | L | L^ | I | T^1 | T^2 | T^3 | λ1 | λ2 |
| C# | 34,869 | 0,229 | 0,5 | 17,434 | 8,443 | 6,906 | 3,874 | 1,835 | 1,835 |
| C++ | 34,869 | 0,332 | 0,5 | 17,434 | 5,818 | 6,906 | 3,874 | 3,865 | 3,865 |

1. Осуществлять циклический сдвиг элементов одномерного массива на заданное число позиций влево.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n\*2 | n1 | n2 | n | N1 | N2 | N | N^ | V\* |
| C# | 3 | 4 | 4 | 8 | 5 | 4 | 9 | 16 | 11,609 |
| C++ | 4 | 4 | 4 | 8 | 5 | 4 | 9 | 16 | 15,509 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V | L | L^ | I | T^1 | T^2 | T^3 | λ1 | λ2 |
| C# | 27 | 0,429 | 0,5 | 13,5 | 3,488 | 5,333 | 3 | 4,991 | 4,991 |
| C++ | 27 | 0,574 | 0,5 | 13,5 | 2,611 | 5,333 | 3 | 8,909 | 8,909 |

1. Заменить все вхождения целочисленного значения в целочисленный массив.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n\*2 | n1 | n2 | n | N1 | N2 | N | N^ | V\* |
| C# | 4 | 4 | 5 | 9 | 4 | 5 | 9 | 19,609 | 15,509 |
| C++ | 5 | 4 | 5 | 9 | 4 | 5 | 9 | 19,609 | 19,651 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | V | L | L^ | I | T^1 | T^2 | T^3 | λ1 | λ2 |
| C# | 28,529 | 0,543 | 0,5 | 14,264 | 2,915 | 6,906 | 3,169 | 8,431 | 8,431 |
| C++ | 28,529 | 0,688 | 0,5 | 14,264 | 2,300 | 6,906 | 3,169 | 13,536 | 13,536 |

1. **Анализ полученных результатов**

Почти во всех задачах многие метрические характеристики совпадают. Отличие кода заключается в том, что на C++ в функцию размер массива приходится передавать отдельным параметром.