Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

Лабораторная работа №4

“Алгоритмы и структуры данных”

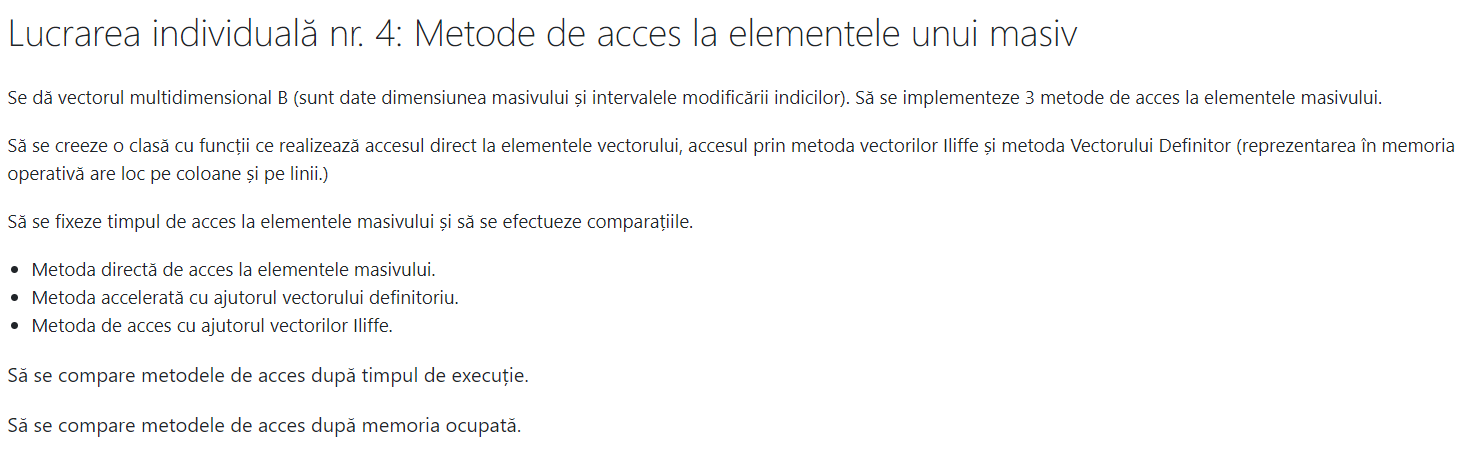
Тема:”Динамические структуры данных”

Проверил: Угнуряну Валерий

Выполнил: Чобану Артём

Группа: i1902

Кишинев 2021

**Задание:**  


<https://github.com/ArtiomCiobanu/Algorithms_Labs>

Код программы:

using System;  
  
namespace Lab4  
{  
 internal class Program  
 {  
 private static void Main()  
 {  
 Range range1 = new Range(-1, 1);  
 Range range2 = new Range(2, 4);  
 Range range3 = new Range(5, 6);  
 Range range4 = new Range(7, 8);  
  
 var array = new Array<int>(range1, range2, range3, range4);  
  
 for (int i = 0; i < array.Length; i++)  
 {  
 array.Elements[i] = i;  
 }  
  
 for (int i1 = range1.L; i1 <= range1.H; i1++)  
 {  
 for (int i2 = range2.L; i2 <= range2.H; i2++)  
 {  
 for (int i3 = range3.L; i3 <= range3.H; i3++)  
 {  
 for (int i4 = range4.L; i4 <= range4.H; i4++)  
 {  
 Console.Write($"[{i1},{i2},{i3},{i4}] = {array.GetValueByDefiningVector(i1, i2, i3, i4)} ");  
 }  
  
 Console.WriteLine();  
 }  
 }  
 }  
  
 int n = 0;  
 for (int i1 = range1.L; i1 <= range1.H; i1++)  
 {  
 for (int i2 = range2.L; i2 <= range2.H; i2++)  
 {  
 for (int i3 = range3.L; i3 <= range3.H; i3++)  
 {  
 for (int i4 = range4.L; i4 <= range4.H; i4++)  
 {  
 array.SetValueByIllifesVector(n, i1, i2, i3, i4);  
 Console.Write($"[{i1},{i2},{i3},{i4}] =

{array.GetValueByIllifesVector(i1, i2, i3, i4)} ");  
 n++;  
 }  
  
 Console.WriteLine();  
 }  
 }  
 }  
  
 Console.WriteLine();  
  
 PrintMeasuredTime(() =>

array.GetRowValue(-1, 2, 5, 7),

"Direct access by rows");  
 PrintMeasuredTime(() =>

array.GetColumnValue(-1, 2, 5, 7),

"Direct access by columns");  
 PrintMeasuredTime(() =>

array.GetValueByDefiningVector(-1,2,5,7),

"Defining vector access time");  
 PrintMeasuredTime(() =>

array.GetValueByIllifesVector(-1, 2, 5, 7),

"Vector Illiffe access");  
 }  
 private static void PrintMeasuredTime(

Func<int> action,

string message)  
 {  
 var beginTime = DateTime.Now;  
 action();  
 var endTime = DateTime.Now;  
 Console.WriteLine($"{message}

{(endTime - beginTime).TotalMilliseconds} ms");  
 }  
 }  
}

namespace Lab4  
{  
 public class Range  
 {  
 public int L { get; }  
 public int H { get; }  
 public int Difference { get; }  
  
 public Range(int l, int h)  
 {  
 L = l;  
 H = h;  
 Difference = h - l;  
 }  
 }  
}

using System.Collections.Generic;  
  
namespace Lab4  
{  
 public class Array<TValue>  
 {  
 private List<int> DefiningVector { get; }  
  
 private List<List<List<List<TValue>>>> IlliffesVector { get; }

public int Length { get; }  
  
 public Range Range1 { get; }  
 public Range Range2 { get; }  
 public Range Range3 { get; }  
 public Range Range4 { get; }  
  
 public TValue[] Elements { get; }  
  
 public Array(Range range1, Range range2, Range range3,

Range range4)  
 {  
 Range1 = range1;  
 Range2 = range2;  
 Range3 = range3;  
 Range4 = range4;  
  
 Length = (range1.Difference + 1) \*  
 (range2.Difference + 1) \*  
 (range3.Difference + 1) \*  
 (range4.Difference + 1);  
 Elements = new TValue[Length];  
  
 DefiningVector = new int[15].ToList();  
 DefiningVector[0] = 4;  
  
 DefiningVector[1] = range1.L;  
 DefiningVector[2] = range1.H;  
 DefiningVector[3] = range2.L;  
 DefiningVector[4] = range2.H;  
 DefiningVector[5] = range3.L;  
 DefiningVector[6] = range3.H;  
 DefiningVector[7] = range4.L;  
 DefiningVector[8] = range4.H;  
  
 DefiningVector[13] = 1;  
 DefiningVector[12] = range4.Difference + 1;  
 DefiningVector[11] = (range3.Difference + 1) \* DefiningVector[12];  
 DefiningVector[10] = (range2.Difference + 1) \* DefiningVector[11];  
  
 DefiningVector[14] = range1.L \* DefiningVector[10] +  
 range2.L \* DefiningVector[11] +  
 range3.L \* DefiningVector[12] +  
 range4.L \* DefiningVector[13];  
  
  
 IlliffesVector = new List<List<List<List<TValue>>>>();  
for (int i1 = range1.L; i1 <= range1.H; i1++)  
{  
 IlliffesVector.Add(new List<List<List<TValue>>>());  
  
 for (int i2 = range2.L; i2 <= range2.H; i2++)  
 {  
 int currenti1 = i1 - range1.L;  
 IlliffesVector[currenti1].Add(new List<List<TValue>>());  
  
 for (int i3 = range3.L; i3 <= range3.H; i3++)  
 {  
 var currenti2 = i2 - range2.L;  
 IlliffesVector[currenti1][currenti2].Add(

new List<TValue>());  
  
 for (int i4 = range4.L; i4 <= range4.H; i4++)  
 {  
 var currenti3 = i3 - range3.L;

IlliffesVector[currenti1][currenti2][currenti3].Add(default);  
 }  
 }  
 }  
}

}  
  
 public void SetValueByIllifesVector(TValue value,

int i1, int i2, int i3, int i4)  
 {  
 IlliffesVector[i1 - Range1.L][i2 - Range2.L]

[i3 - Range3.L][i4 - Range4.L] = value;

}  
  
 public TValue GetValueByIllifesVector(

int i1, int i2, int i3, int i4)  
 {  
 return IlliffesVector[i1 - Range1.L][i2 - Range2.L]

[i3 - Range3.L][i4 - Range4.L];

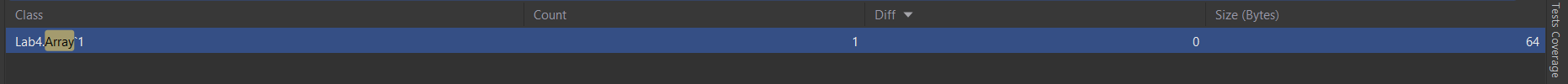
public TValue GetRowValue(int i1, int i2, int i3, int i4)  
 {  
 int d3 = Range4.Difference + 1;  
 int d2 = (Range3.Difference + 1) \* d3;  
 int d1 = (Range2.Difference + 1) \* d2;  
  
 int finalIndex = i1 - Range1.L \* d1 +  
 (i2 - Range2.L) \* d2 +  
 (i3 - Range3.L) \* d3 +  
 (i4 - Range4.L);  
  
 return Elements[finalIndex];  
 }  
  
 public TValue GetColumnValue(int i1, int i2, int i3, int i4)  
 {  
 int d2 = Range1.Difference + 1;  
 int d3 = (Range2.Difference + 1) \* d2;  
 int d4 = (Range3.Difference + 1) \* d3;  
  
 int finalIndex = i1 - Range1.L +  
 (i2 - Range2.L) \* d2 +  
 (i3 - Range3.L) \* d3 +  
 (i4 - Range4.L) \* d4;  
  
 return Elements[finalIndex];  
 }

public TValue GetValueByDefiningVector(

int i1, int i2, int i3, int i4)  
 {  
 int finalIndex = i1 \* DefiningVector[10] +  
 i2 \* DefiningVector[11] +  
 i3 \* DefiningVector[12] +  
 i4 \* DefiningVector[13]  
 - DefiningVector[14];  
  
 return Elements[finalIndex];  
 }  
 }  
}

Анализ использования памяти:

Прямой доступ – 64 байт:



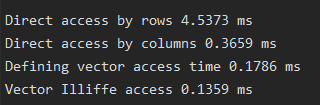
Определяющий вектор – 40 байт:



Вектор Иллиффе – 32 байта:



Анализ скорости доступа:



Как видно, доступ с помощью вектора Иллиффе быстрее остальных.