ГУ ВПО «Белорусско-Российский университет»

Кафедра «Автоматизированные системы управления»

Лабораторная работа № 3

**Двумерные массивы**

по дисциплине «Типы и структуры данных»

Вариант 6, 13

Выполнил студент гр. ПМР-211

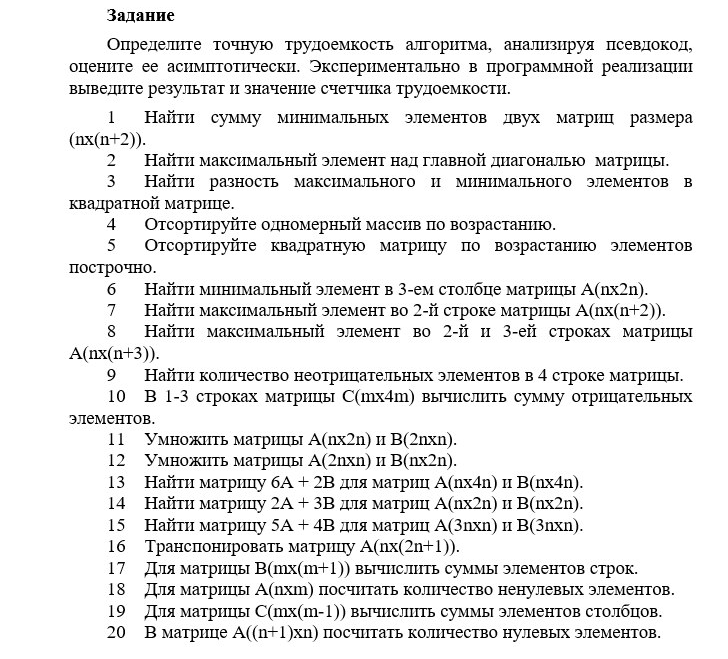
Азарёнок А.Р.а ааааааааааааааа

Проверила ст.преподаватель ааа

Беккер И. А. аааааааааааааааааа.

Могилев 2023

Цель работы: определить точную трудоемкость алгоритма, анализируя псевдокод, оценить асимптотически; экспериментально в программной реализации вывести результат и значение счетчика трудоемкости.



Листинг программы:

using System;

using System.Threading;

namespace lab3

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите n: ");

int n = Convert.ToInt16(Console.ReadLine());

Console.WriteLine();

int count1 = 0;

int count2 = 0;

// 6) найти минимальный элемент в 3-ем столбце матрицы A(n\*2n)

Console.WriteLine("A:");

int[,] A = CreateMatrix(n, 2 \* n);

int min = A[0, 2];

ShowMatrix(A);

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

{

min = min > A[i, 2] ? A[i, 2] : min;

count1++;

}

Console.WriteLine($"Минимальный элемент в 3-ем столбце: {min}\nТрудоемкость = {count1}\n");

// 13) найти матрицу 6B + 2C для матриц B(n\*4n) и C(n\*4n)

Console.WriteLine("B:");

int[,] B = CreateMatrix(n, 4 \* n);

ShowMatrix(B);

Console.WriteLine("C:");

int[,] C = CreateMatrix(n, 4 \* n);

ShowMatrix(C);

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

{

for (int j = 0; j < 4 \* n; j++)

{

B[i, j] = 6\*B[i, j] + 2\*C[i, j];

count2++;

}

}

Console.WriteLine($"Матрица 6B + 2C:");

ShowMatrix(B, 4);

Console.WriteLine($"Трудоемкость = {count2}");

}

static int[,] CreateMatrix(int n, int m)

{

Random rnd = new Random();

int[,] matrix = new int[n, m];

Thread.Sleep(10);

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

{

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

{

matrix[i, j] = rnd.Next(50);

}

}

return matrix;

}

static void ShowMatrix(int[,] matrix, int pad = 3)

{

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < matrix.GetLength(1); j++)

{

Console.Write(matrix[i, j].ToString().PadLeft(pad));

}

Console.WriteLine();

}

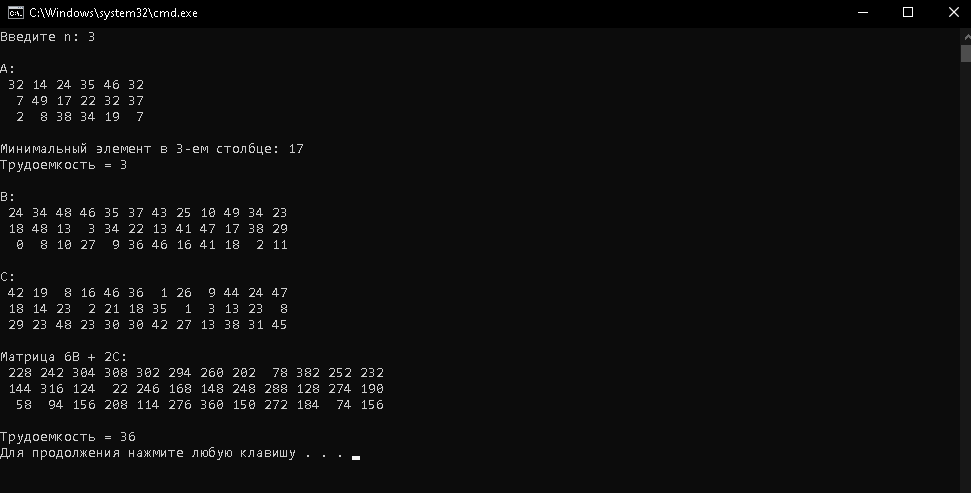
Console.WriteLine();

}

}

}

Вывод программы:



Вывод: экспериментально доказано что поиск по столбцам матрицы размера n\*2n требует O(n); также для суммы двух матриц размера n\*4n необходимо O(4n^2), что эквивалентно O(n^2).