ФГОУ ВПО Кубанский государственный технологический университет

(КубГТУ)

Кафедра Информационных систем и программирования

Факультет Компьютерных технологий и автоматизированных систем

**КУРСОВАЯ РАБОТА**

По дисциплине Дискретная математика

На тему Задачи и алгоритмы дискретной математики.

Выполнил студент группы 12-КБ-ПИ1 Мацко Сергей Александрович

Допущен(а) к защите :

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.А.Симоненко

(подпись, дата, расшифровка подписи)

Защищён \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Е.А.Симоненко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Г. Волик

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, расшифровка подписи)

Краснодар

2013

ФГОУ ВПО Кубанский государственный технологический университет

(КубГТУ)

Кафедра Информационных систем и программирования

Факультет Компьютерных технологий и автоматизированных систем

УТВЕРЖДАЮ

Зав. кафедрой Видовский Л.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись, расшифровка подписи)

**ЗАДАНИЕ**

на курсовую работу

Студенту Мацко Сергею Александровичу группы 12-КБ-ПИ1

факультета Компьютерных технологий и автоматизированных систем

специальности 231000 – Программная инженерия

Тема работы Задачи и алгоритмы дискретной математики.

Содержание задания: Изучить темы «Планарный граф. Проверка графа на планарность» и «Нахождения кратчайшего пути. Алгоритм Флойда», провести исследования алгоритмов работы с этими объектами, составить программы, которые демонстрируют указанные алгоритмы, провести тестирование программ.

Объём курсовой работы:

а) пояснительная записка стр. 17;

Рекомендуемая литература Уайс «Алгоритмы», Скиена «Алгоритмы».

Срок выполнения курсовой: с 25 сентября 2013г. по 28 декабря 2013г.

Срок защиты: с 25 сентября 2013г. по 28 декабря 2013г.

Дата выдачи задания: с 25 сентября 2013г. по 28 декабря 2013г.

Дата сдачи работы на кафедру: с 25 сентября 2013г. по 28 декабря 2013г.

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись)

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Задание принял студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

РЕФЕРАТ

Система непересекающихся множеств, Кратчайшие пути: алгоритм Дейкстры

Стр. 11, табл. 2 , рис. 4, библ. 1.

Содержание

[Введение 5](#_Toc375600953)

[1. Глава Алгоритм Флойда 6](#_Toc375600954)

[2. Глава планарность. Проверка графа на планарность. 7](#_Toc375600955)

[Список используемых источников 9](#_Toc375600956)

# Введение

В данной курсовой работе рассмотрены два вопросы. Первый из них –Алгоритм Флойда. Второй – Проверка графа на планарность.

# 

# ****Глава**** Алгоритм Флойда

Постановка задачи:

Дан ориентированный граф с n вершинами и m ребрами ,длины всех ребер не отрицательные , нужно найти минимальный путь от одной вершины к другой.

Алгоритм:

1)создаем изначальную матрицу длин D0 n\*n(где бесконечность обозначается 1000). Инициализируем k=n.

2) проверяем каждый D[i;j] элемент таблицы , если он больше D[i;k]+D[k;j] то выполняем замену D[i;j] на D[i;k]+D[k;j] и так пока k меньше n

3)выводим конечную матрицу длин на экран.

Входные данные:

Дан граф(рисунок.1) по графу составить матрицу длин D0

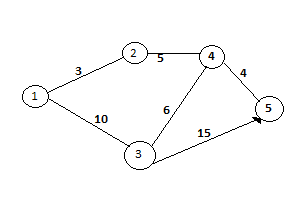


Рисунок 1.Направленный граф.

D0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | - | 3 | 10 | 1000 | 1000 |
| 2 | 3 | - | 1000 | 5 | 1000 |
| 3 | 10 | 1000 | - | 6 | 15 |
| 4 | 1000 | 5 | 6 | - | 4 |
| 5 | 1000 | 1000 | 1000 | 4 | - |

Результат работы программы:

Конечная матрица длин D4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | - | 3 | 10 | 8 | 12 |
| 2 | 3 | - | 11 | 5 | 9 |
| 3 | 10 | 11 | - | 6 | 10 |
| 4 | 8 | 5 | 6 | - | 4 |
| 5 | 12 | 9 | 10 | 4 | - |

Реализация алгоритма в Приложении А.

# Глава планарность. Проверка графа на планарность.

Задача 1.

Постановка задачи:

Проверить граф (рисунок 2) на планарность.

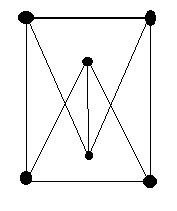


Рисунок 2. граф

1)Считаем количество вершин (n),ребер(m) и граней( r), в данном графе n=6, m= 9, r=5

2) Проверяем граф на необходимое условие планарности: отсутствие подграфов К5 и К3,3 ,то-есть пытаемся уложить граф на плоскость (рисунок 3)

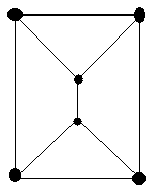


Рисунок 3 граф в плоскости

3)Проверяем граф на достаточное условие планарности n-m+r=2, в данном графе: 6-9+5=2.граф планарен

4)Выход из программы

Задача 2.

Постановка задачи:

Проверить граф (рисунок 4) на планарность.

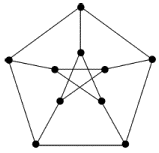


Рисунок 4 граф

1) Считаем количество вершин (n),ребер(m) и граней( r), в данном графе n=10, m= 15, r=6

2) Проверяем граф на необходимое условие планарности: отсутствие подграфов К5 и К3,3, данный граф содержит подграф К5 по данному условию граф не подходит под категорию планарных графов.

3)чтоб удостовериться в том что граф не планарен , проверим его на достаточное условие:10-15+6<2 это доказывает , что граф не планарен.

4) выход из программы.

# Список используемых источников

1. [Скиена] Скиена С. Алгоритмы. Руководство по разработке. – 2-е изд.: пер. с англ. – СПб.: БХВ-Петербург, 2011. – 720 с. Скиена «Алгоритмы».
2. [Седжвик] Седжвик Р. Алгоритмы на C++. – Пер. с англ. – М.: Вильямс, 2011. – 1056 с.
3. [Окулов: ДМ] Окулов С.М. Дискретная математика. Теория и практика решения задач по информатике: учебное пособие. – М.: БИНОМ. Лаборатория знаний, 2008. – 422 с.

**Приложение А**

**Исходный код –**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Алгоритм\_Флойда

{

class Program

{

static void Main()

{

int[,] LNG = new int[5, 5] // задаем матрицу длин между вершинами (1000-обозначен путь между не соединенными вершинами)

{{0,3,10,1000,1000},

{3,0,1000,5,1000},

{10,1000,0,6,15},

{1000,5,6,0,4},

{1000,1000,1000,4,0}};

int i, j, k; // инициализация переменных

for (k = 0; k < 5; k++)

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

for (j = 0; j < 5; j++)

if (LNG[i, j] > LNG[i, k] + LNG[k, j])

LNG[i, j] = LNG[i, k] + LNG[k, j];

// выведение конечной матрицы длин

Console.WriteLine("Алгоритм Флойда : ");

Console.WriteLine(" 1 2 3 4 5 ");

Console.WriteLine(" \_\_\_\_\_\_\_\_\_\_\_\_");

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

{

Console.Write((i + 1) + "| ");

for (j = 0; j < 5; j++)

Console.Write("{0} ", LNG[i, j]);

Console.WriteLine("\n |");

}

Console.ReadLine();

}

}

}