9Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №9**

**Дисциплина: Обработка больших данных**

Работу выполнила: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. Н. Баева

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. И. Шиян

**Цель:** научиться работать с графами. Освоить основные функции обработки графов – их задание, визуализация, оформление, извлечение информации о графе.

**Ход работы:**

Сперва была установлена библиотека igraph, необходимая для работы с графами. Далее была создана переменная N, которая отвечает за количество вершин (согласно варианту).

Задание 1.

1. Создадим кольцевой граф g со случайным числом вершин G\_size (от N+10 до (N/10+5)2+5N). Выведем число ребер и вершин этого графа (рисунок 1). Построим граф (рисунок 2), выведем его матрицу смежности (рисунок 3).

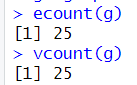


Рисунок 1 – Число ребер и вершин графа g

Изображение выглядит как круг, дизайн

Автоматически созданное описание

Рисунок 2 – Кольцевой граф g

Изображение выглядит как текст, снимок экрана, линия, График

Автоматически созданное описание

Рисунок 3 – Матрица смежности графа g

2. Создадим граф g1 из пустого графа с числом вершин G\_size желтого цвета. Добавим ему N\*8 случайных ребер, сформированных из вектора вершин, окрасим ребра красным цветом, нарисуем граф (рисунок 4). Добавим графу g1 еще N\*10 случайных ребер, сформированных из вектора вершин, окрасим ребра синим цветом, нарисуем граф (рисунок 5).

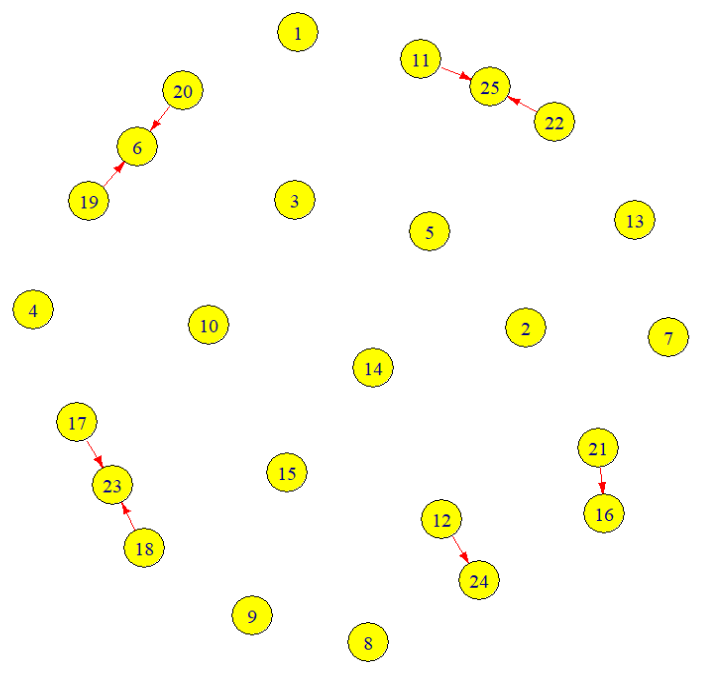


Рисунок 4 – Граф первого варианта

Изображение выглядит как круг, диаграмма

Автоматически созданное описание

Рисунок 5 – Граф второго варианта

3. Добавим ребра между вершиной 2N+23 и 2N+20, 2N+12 и N+15, 2N-1 и N+8, 2N и 2N+1, N+7 и N+13, окрасим их в черный цвет. Нарисуем граф (рисунок 6). Выведем соседей N-й вершины (рисунок 7), ребра, инцидентные этой вершине (рисунок 8). Выясним, соединены ли вершины N+10 и N+12 (рисунок 9). Выведем матрицу смежности (рисунок 10).

Изображение выглядит как круг, диаграмма

Автоматически созданное описание

Рисунок 6 – Граф с чёрными рёбрами

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Рисунок 7 – Соседи N-й вершины

Изображение выглядит как текст, Шрифт, белый, снимок экрана

Автоматически созданное описание

Рисунок 8 – Ребра, инцидентные N-ой вершине

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Рисунок 9 – Проверка на соединение вершин N+10 и N+12

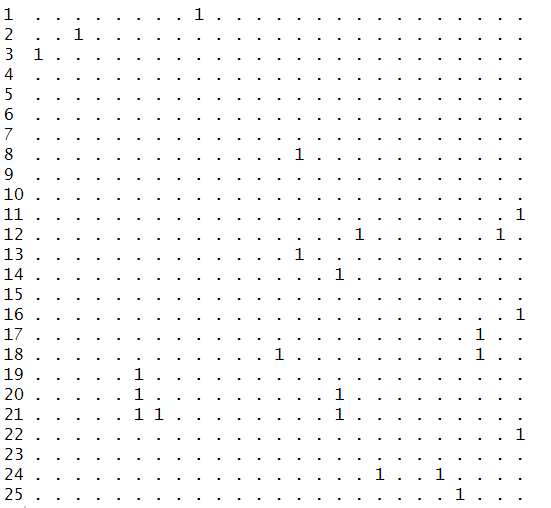


Рисунок 10 – Матрица смежности g

4. Добавим еще одну вершину и подключим ее к той, которая имеет наибольшее количество связанных с ней узлов. Присвоим имена всем вершинам (например, буквы в алфавитном порядке – используем заглавные и, если не хватит, строчные буквы). Построим граф (рисунок 11). Выведем матрицу смежности (рисунок 12). Выберем вершины, для которых значение связности меньше 5 и больше 2 (рисунок 13).

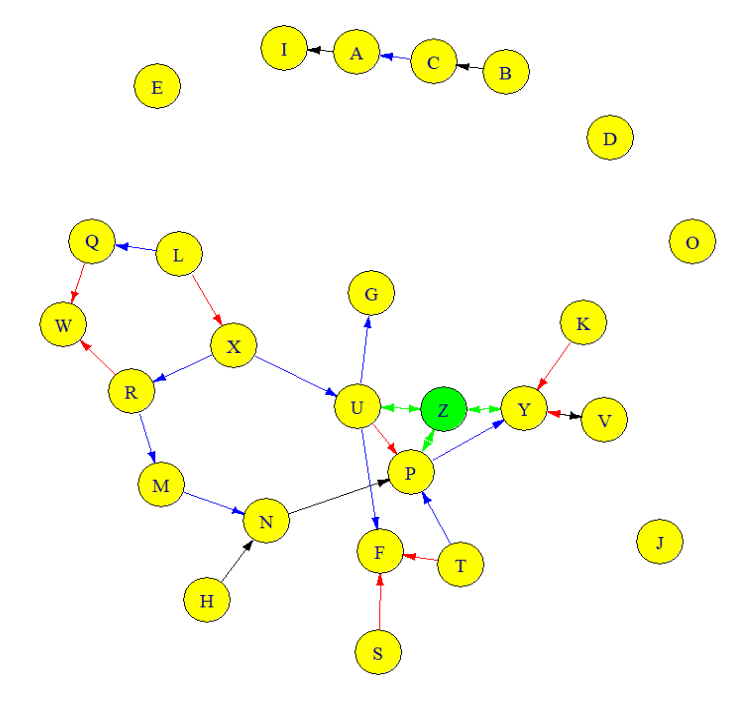


Рисунок 11 – Граф с новой вершиной

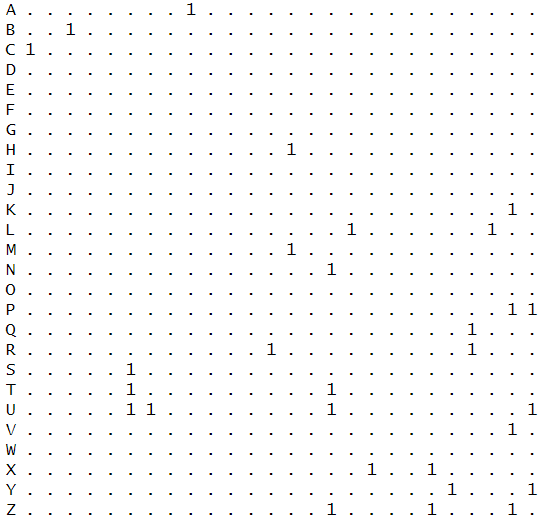


Рисунок 12 – Матрица смежности g



Рисунок 13 – Вершины, значение связности меньше 5 и больше 2

5. Испробуем алгоритмы размещения графа (in\_circle (рисунок 14), in\_tree (рисунок 15), lattice (рисунок 16)).

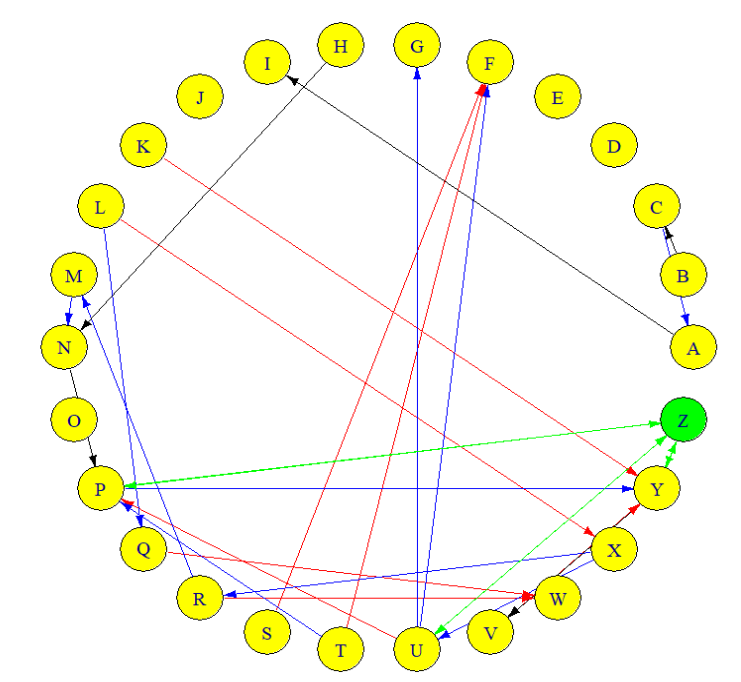


Рисунок 14 – Размещение графа как in\_circle

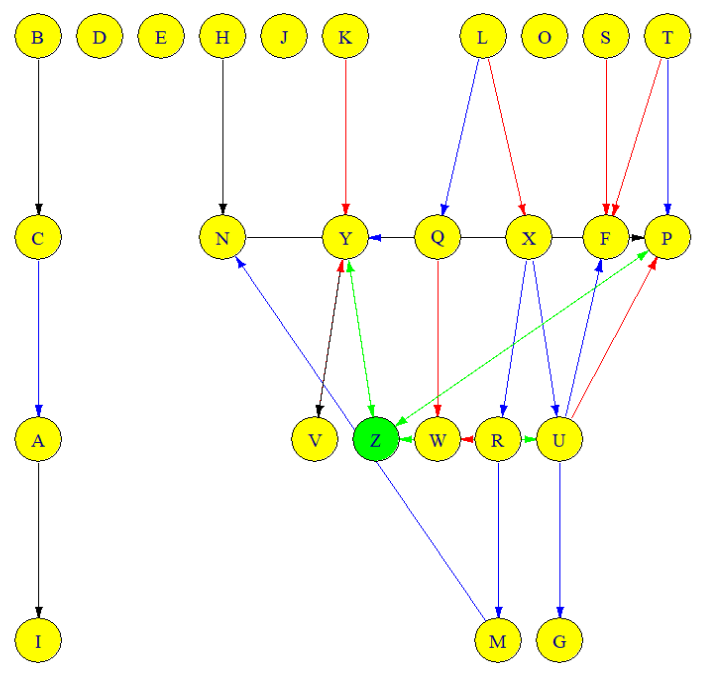


Рисунок 15 – Размещение графа как in\_tree

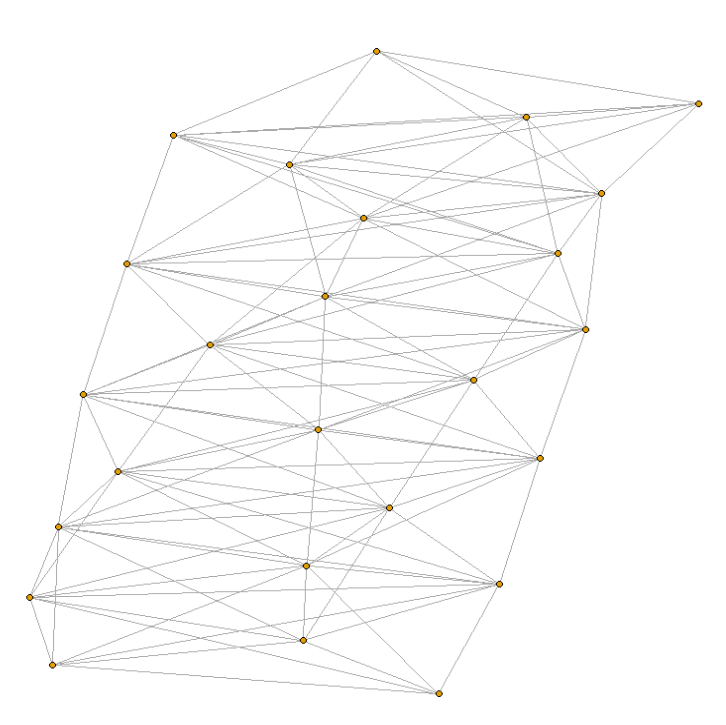


Рисунок 16 – Размещение графа как lattice

6. Выполним измерение диаметра графа g1 (рисунок 17), выведем список самых коротких путей для каждой вершины (рисунок 18) и откалибруем величины вершин согласно их степеней (рисунок 19).



Рисунок 17 – Диаметр графа g1

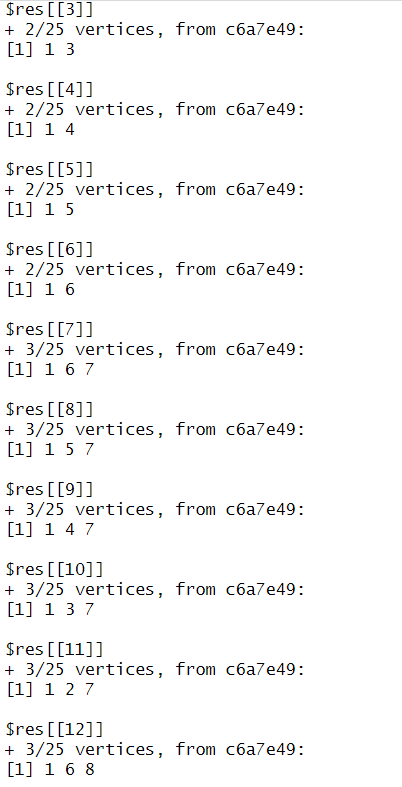


Рисунок 18 – Часть списка самых коротких путей для каждой вершины

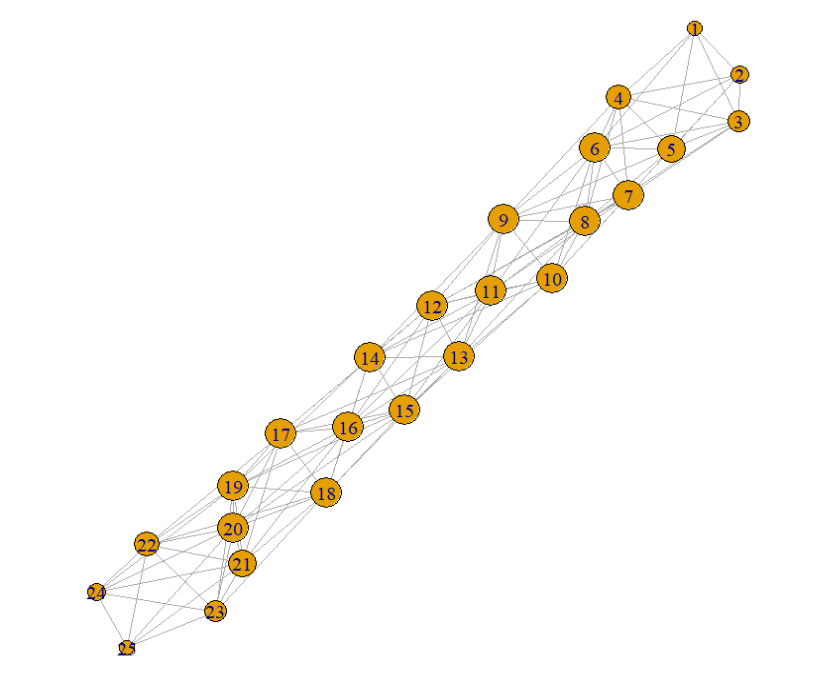


Рисунок 19 – Откалиброванный граф

Задание 2.

5. В Банановой республике очень много холмов, соединенных мостами. На химическом заводе произошла авария, в результате чего испарилось экспериментальное удобрение "зован". На следующий день выпал цветной дождь, причем он прошел только над холмами. В некоторых местах падали красные капли, в некоторых - синие, а в остальных - зеленые, в результате чего холмы стали соответствующего цвета. Президенту Банановой республики это понравилось, но ему захотелось покрасить мосты между вершинами холмов так, чтобы мосты были покрашены в цвет холмов, которые они соединяют. К сожалению, если холмы разного цвета, то покрасить мост таким образом не удастся. Посчитайте количество таких "плохих" мостов.

Формат входных данных:

Число холмов N (20 <= N <= 200) необходимо ввести с клавиатуры. Далее сформируйте матрицу смежности, описывающую наличие мостов между холмами (1-мост есть, 0-нет). Сформируйте вектор "color", содержащий N чисел, обозначающих цвет холмов: 1 - красный; 2 - синий; 3 – зеленый и т. п.

Формат выходных данных:

На выходе нужно получить количество "плохих" мостов и номера холмов "плохих" мостов.

Алгоритм решения:

Введем число холмов N. Сформируем матрицу смежности, описывающую наличие мостов между холмами (1-мост есть, 0-нет). Сформируем вектор "color", содержащий N чисел, обозначающих цвет холмов: 1 - красный; 2 - синий; 3 – зеленый и т. п. (рисунок 20).

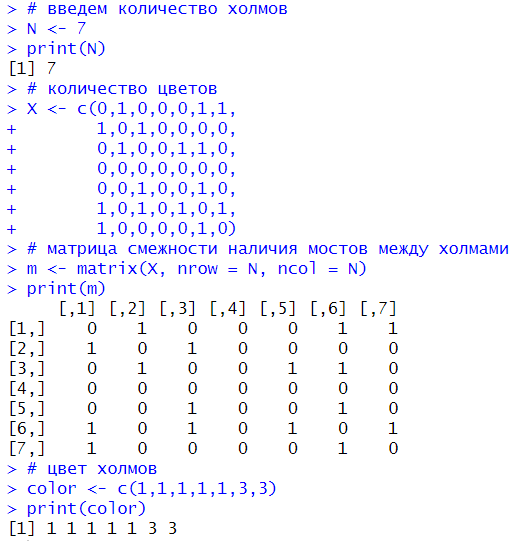


Рисунок 20 – Данные для задачи

Введенная матрица является матрицей смежности. Создадим неориентированный граф (рисунок 21).

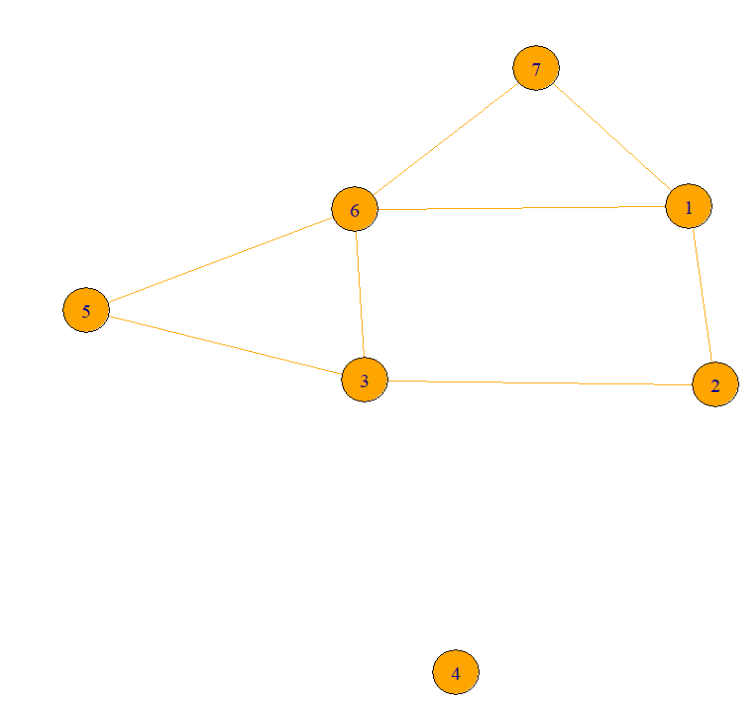


Рисунок 21 – Неориентированный граф

Раскрасим вершины графа в красный, синий и зеленый цвета (рисунок 22).

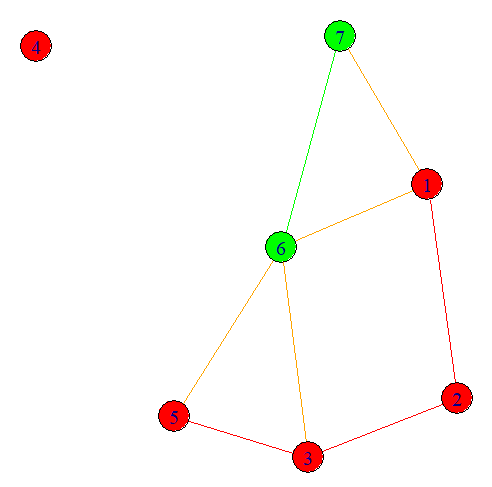


Рисунок 22 – Граф с раскрашенными ребрами

Покрасим ребра между вершинами так, чтобы ребра были покрашены в цвет вершин, которые они соединяют (рисунок 23).

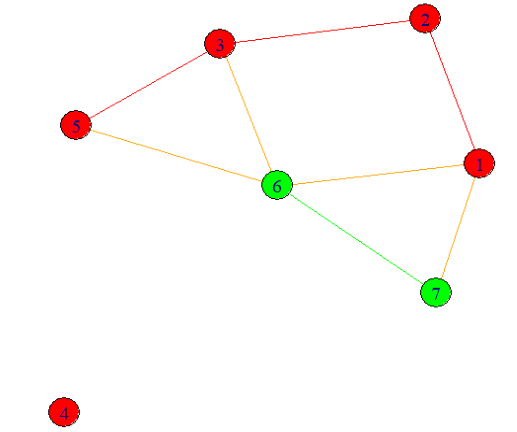


Рисунок 23 – Граф с раскрашенными ребрами

Выведем количество "плохих" мостов и номера холмов "плохих" мостов:

(рисунок 24).

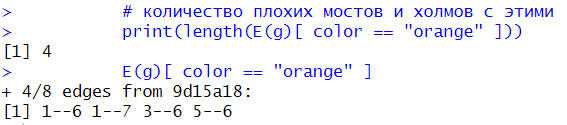


Рисунок 24 – Результат

**Вывод:** В данной работе были ознакомлены с работой с графами. Были освоены основные функции обработки графов – их задание, визуализация, оформление, извлечение информации о графе.

**ПРИЛОЖЕНИЕ А**

**Листинг программы с комментариями**

# скачивание необходимой библиотеки

library(igraph)

# номер варианта

N<-1

################################################################################

# Задание 1.

# 1. Создайте кольцевой граф g со случайным числом вершин G\_size (от N+10 до (N/10+5)^2+5N).

# Выведите число ребер и вершин этого графа. Постройте граф, выведите его матрицу смежности.

G\_size<-sample((N+10):((N%/%10+5)\*\*2+5\*N), 1)

g<-graph.ring(n=G\_size)

ecount(g)

vcount(g)

plot(g, edge.arrow.size=.2, vertex.size=13)

g[]

# 2. Создайте граф g1 из пустого графа с числом вершин G\_size желтого цвета.

# Добавьте ему 8N случайных ребер, сформированных из вектора вершин, окрасьте ребра красным цветом, нарисуйте граф и выведите его матрицу смежности.

# Добавьте графу g1 еще 10N случайных ребер, сформированных из вектора вершин, окрасьте ребра синим цветом, нарисуйте граф и выведите его матрицу смежности.

g1<-graph.empty()+vertices(1:G\_size, color='yellow')

g1<-g1+edges(sample(V(g1), 2\*8\*N, replace=TRUE), color='red')

plot(g1, edge.arrow.size=.2, vertex.size=13)

g1[]

g1<-g1+edges(sample(V(g1), 2\*10\*N, replace=TRUE), color='blue')

plot(g1, edge.arrow.size=.2, vertex.size=13)

g1[]

# 3. Добавьте ребра между вершиной 2N+23 и 2N+20, 2N+12 и N+15, 2N-1 и N+8, 2N и 2N+1, N+7 и N+13, окрасьте их в черный цвет (предварительно проверьте существуют ли такие вершины – функцией %in% либо match, для несуществующих вершин ребра не добавляйте).

# Нарисуйте граф. Выведите соседей N-й вершины, ребра, инцидентные этой вершине.

# Соединены ли вершины N+10 и N+12? Выведите матрицу смежности.

v<-c(2\*N+23, 2\*N+20, 2\*N+12, N+15, 2\*N-1, N+8, 2\*N, 2\*N+1, N+7, N+13)

for (i in seq(1, length(v), 2)) {

if (v[i] %in% V(g1) && v[i+1] %in% V(g1)) {

g1<-add.edges(g1, c(v[i],v[i+1]), color='black')

}

}

plot(g1, edge.arrow.size=.2, vertex.size=13)

neighbors(g1, V(g1)[N], mode='out')

incident(g1, V(g1)[N], mode='all')

if ((N+10) %in% V(g1) && (N+12) %in% V(g1)) {

are.connected(g1, V(g1)[N+10], V(g1)[N+12])

}

g1[]

# 4. Добавьте еще одну вершину и подключите ее к той, которая имеет наибольшее количество связанных с ней узлов.

# Присвойте имена всем вершинам (например, буквы в алфавитном порядке – используйте заглавные и, если не хватит, строчные буквы).

# Выведите матрицу смежности. Выберите вершины, для которых значение связности меньше 5 и больше 2.

x<-length(V(g1))+1

g1<-g1+vertices(x, color='green')

deg<-degree(g1, mode='all')

for (i in which(deg==max(deg))) {

g1<-g1+edges(c(x,i, i,x), color='green')

}

v<-c(toupper(letters[1:26]), tolower(letters[1:26]))

mn<-min(length(V(g1)), length(v))

g1<-set\_vertex\_attr(g1, 'name', 1:mn, v[1:mn])

plot(g1, edge.arrow.size=.2, vertex.size1=13)

g1[]

v<-names(V(g1))

deg<-degree(g1, mode='all')

v[which(deg<5&deg>2)]

# 5. Испробуйте алгоритмы размещения Вашего графа (in\_circle, as\_tree, lattice).

coords<-layout\_(g1, in\_circle())

plot(g1, layout=coords, edge.arrow.size=.2)

coords<-layout\_(g1, as\_tree())

plot(g1, layout=coords, edge.arrow.size=.2)

g1<-graph.lattice(length=25,dim=1,nei=5, circular=FALSE)

plot(g1,vertex.size=2,vertex.label=NA,layout=layout.kamada.kawai,edge.arrow.size=.1)

# 6. Выполните измерение диаметра графа g1,

# выведите список самых коротких путей для каждой вершины и откалибруйте величины вершин согласно их степеней.

diameter(g1)

all\_shortest\_paths(g1, 1, to=V(g1), mode='all', weights=NULL)

deg<-degree(g1, mode='all')

plot(g1, edge.arrow.size=.2, vertex.size=deg)

################################################################################

################################################################################

# Задание 2.

# 5 задание.

#В Банановой республике очень много холмов, соединенных мостами. На химическом заводе

#произошла авария, в результате чего испарилось экспериментальное удобрение "зован". На

#следующий день выпал цветной дождь, причем он прошел только над холмами. В некоторых местах

#падали красные капли, в некоторых - синие, а в остальных - зеленые, в результате чего холмы

#стали соответствующего цвета. Президенту Банановой республики это понравилось, но ему захотелось

#покрасить мосты между вершинами холмов так, чтобы мосты были покрашены в цвет холмов, которые они

#соединяют. К сожалению, если холмы разного цвета, то покрасить мост таким образом не удастся.

#Посчитайте количество таких "плохих" мостов.

# введем количество холмов

N <- 7

print(N)

# количество цветов

X <- c(0,1,0,0,0,1,1,

1,0,1,0,0,0,0,

0,1,0,0,1,1,0,

0,0,0,0,0,0,0,

0,0,1,0,0,1,0,

1,0,1,0,1,0,1,

1,0,0,0,0,1,0)

# матрица смежности наличия мостов между холмами

m <- matrix(X, nrow = N, ncol = N)

print(m)

# цвет холмов

color <- c(1,1,1,1,1,3,3)

print(color)

# создание неориентированного графа

g <- graph.adjacency(m, mode = "undirected")

V(g)$color <- "orange"

E(g)$color <- "orange"

plot(g)

# раскраска вершин

V(g)$color <- c("red", "blue", "green")[color]

plot(g)

# раскраска ребер

red <- V(g)[ color == "red" ]

blue <- V(g)[ color == "blue" ]

green <- V(g)[ color == "green" ]

E(g)[ red %--% red ]$color <- "red"

E(g)[ blue %--% blue ]$color <- "blue"

E(g)[ green %--% green ]$color <- "green"

plot(g)

# количество плохих мостов и холмов с этими

print(length(E(g)[ color == "orange" ]))

E(g)[ color == "orange" ]