МИНИСТЕРТСВО 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**

**ОТЧЕТ**

**о выполнении лабораторной работы №9**

по дисциплине «Обработка больших данных»

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ студент 39/2 группы, Мандыч Д.И.

Работу проверил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_преподаватель Шиян В.И.

Краснодар 2023

Исходный код: <https://github.com/SnowLukin/BigData/tree/main>

**Задание 1**

A picture containing fashion accessory, circle, bead, jewellery

Description automatically generated

Рисунок 1 – Кольцевой граф, с количеством вершин, полученным по

заданной формуле

A picture containing text, screenshot

Description automatically generated

Рисунок 2 – Фрагмент матрицы смежности кольцевого графа

A picture containing circle

Description automatically generated

Рисунок 3 – Добавление красных ребер в пустой граф

A picture containing circle, colorfulness

Description automatically generated

Рисунок 4 – Добавление голубых ребер в предыдущий граф

A screenshot of a computer program

Description automatically generated with low confidence

Рисунок 5 – Фрагмент матрицы смежности полученного графа

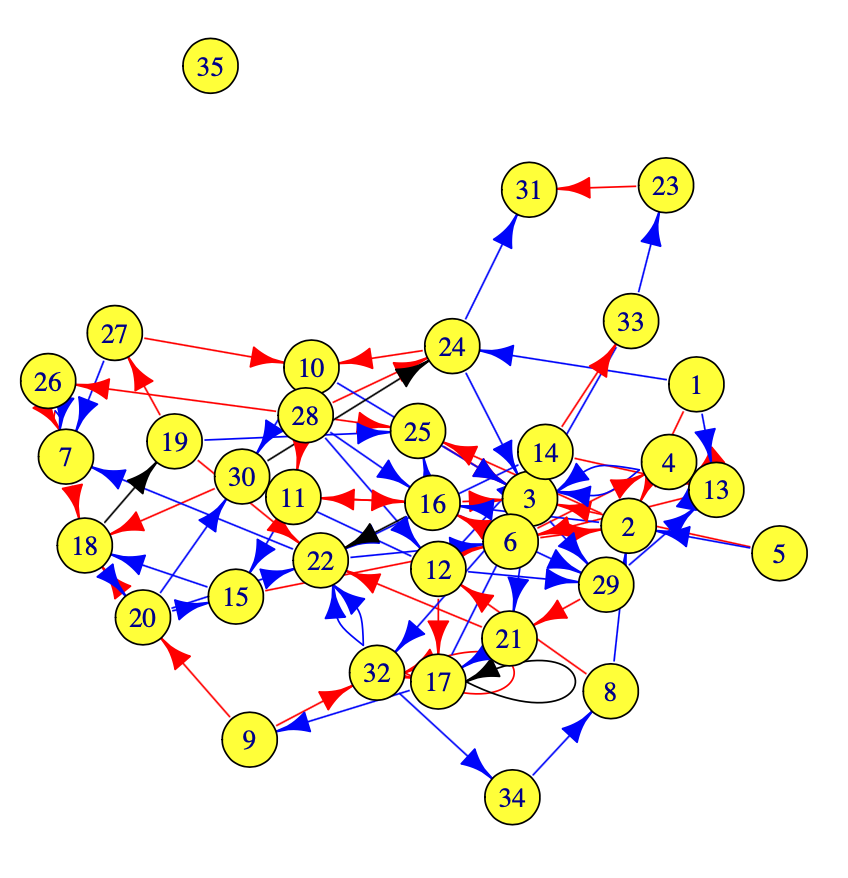


Рисунок 6 – Фрагмент матрицы смежности полученного графа

A black background with white text

Description automatically generated with low confidence

Рисунок 7 – Соседи инцидентные ребра N-ной вершины. Проверка на наличие указанного ребра

A picture containing text, screenshot, black

Description automatically generated

Рисунок 8 – Фрагмент матрицы смежности предыдущего графа

A picture containing circle, symmetry, art, colorfulness

Description automatically generated

Рисунок 9 – Кольцевое представление полученного графа

A picture containing line, circle, diagram, screenshot

Description automatically generated

Рисунок 10 – Древовидное представление полученного графа

A picture containing circle, line, diagram, colorfulness

Description automatically generated

Рисунок 11 – Сетчатое представление графа

A screenshot of a computer

Description automatically generated with medium confidence

Рисунок 12 – Диаметр графа и кратчайшие пути

**Задание 2**

Вводится N - количество домов и К - количество дорог. Дома пронумерованы от 1 до N. Каждая дорога определяется тройкой чисел - двумя номерами домов - концов дороги и длиной дороги. В каждом доме живет по одному человеку. Найти точку - место встречи всех людей, от которой суммарное расстояние до всех домов будет минимальным. Указать номер дома. Примечание: длины дорог – положительные целые числа.

Код:

*# Dijkstra's algorithm function*dijkstra <- *function*(graph, source) {  
 n <- *nrow*(graph)  
 dist <- *rep*(1e9, n)  
 dist[source] <- 0  
 visited <- *rep*(FALSE, n)  
  
 *for* (i *in* 1:n) {  
 u <- -1  
  
 *for* (j *in* 1:n) {  
 *if* (!visited[j] && (u == -1 || dist[j] < dist[u])) {  
 u <- j  
 }  
 }  
  
 visited[u] <- TRUE  
  
 *for* (v *in* 1:n) {  
 *if* (!visited[v] && graph[u, v] < 1e9) {  
 new\_dist <- dist[u] + graph[u, v]  
 *if* (new\_dist < dist[v]) {  
 dist[v] <- new\_dist  
 }  
 }  
 }  
 }  
  
 *return*(dist)  
}  
  
*# Read input  
# N <- as.integer(readline(prompt = "Enter the number of houses (N): "))  
# K <- as.integer(readline(prompt = "Enter the number of roads (K): ")) <-*N <- 6 *# Number of houses*K <- 7 *# Number of roads  
  
# Initialize distance matrix with large values*graph <- *matrix*(1e9, nrow = N, ncol = N)  
  
*# Roads: (A, B, length)*roads <- *list*(  
 *c*(1, 2, 4),  
 *c*(1, 3, 2),  
 *c*(2, 3, 1),  
 *c*(2, 4, 5),  
 *c*(3, 4, 8),  
 *c*(3, 5, 10),  
 *c*(4, 6, 6)  
)  
  
*# Read roads and update graph  
for* (i *in* 1:K) {  
 *# road <- as.integer(unlist(strsplit(readline(prompt = "Enter road details (A, B, length): "), " ")))* road <- roads[[i]]  
 graph[road[1], road[2]] <- road[3]  
 graph[road[2], road[1]] <- road[3]  
}  
  
*# Calculate all-pairs shortest paths using Dijkstra's algorithm*all\_pair\_shortest\_paths <- *lapply*(1:N, *function*(source) *dijkstra*(graph, source))  
dist\_matrix <- *do.call*(rbind, all\_pair\_shortest\_paths)  
  
*# Find the house with minimum sum of distances*min\_distance <- *sum*(dist\_matrix[1, ])  
meeting\_point <- 1  
  
*for* (i *in* 2:N) {  
 curr\_distance <- *sum*(dist\_matrix[i, ])  
 *if* (curr\_distance < min\_distance) {  
 min\_distance <- curr\_distance  
 meeting\_point <- i  
 }  
}  
  
*cat*("The optimal meeting point is house number", meeting\_point, "\n")

Пояснение:

Задача решается с помощью алгоритма Дейкстры, который используется для нахождения кратчайшего пути между вершинами графа. В этом случае граф представляет собой сеть домов и дорог между ними. Ниже представлена функция dijkstra, которая принимает граф и начальную вершину и возвращает список кратчайших расстояний от этой вершины до всех остальных.

Затем считывается количество домов N и количество дорог K. В данном примере мы устанавливаем значения N = 6 и K = 7.

Мы инициализируем матрицу расстояний graph размером NxN с большими значениями (1e9). Затем мы определяем список roads, который содержит информацию о дорогах и их длинах.

Далее мы заполняем матрицу расстояний graph информацией из списка roads. Если между домами A и B есть дорога длиной L, то graph[A, B] и graph[B, A] присваивается значение L.

Теперь для каждого дома мы применяем алгоритм Дейкстры, чтобы найти кратчайшие пути до всех других домов. Мы сохраняем результаты в матрице dist\_matrix.

Наконец, мы ищем дом с минимальной суммой расстояний до всех других домов. Мы начинаем с первого дома и обновляем значение min\_distance и номер дома meeting\_point при нахождении дома с меньшей суммой расстояний.

Выводим на экран номер дома, который является оптимальной точкой встречи.

Таким образом, программа находит дом, от которого суммарное расстояние до всех остальных домов минимально, и выводит его номер. В данном примере оптимальной точкой встречи является дом номер 2.

**Вывод**

Научились работать с графами. Освоили основные функции обработки графов – их задание, визуализация, оформление, извлечение информации о графе.