

Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра «Вычислительная техника»

**Отчёт**  
по лабораторной работе №9  
по курсу “Логика и основы алгоритмизации в инженерных задачах”  
на тему “Поиск расстояний в графе”

Выполнили:

Студенты группы 24ВВВ4  
Суходолов И.А.  
Чернышевский Е.И.

Приняли:

к.т.н., доцент Юрова О.В.

к.э.н., доцент Акифьев И.В.

Пенза 2025

**Название:**

Поиск расстояний в графе

**Цель работы:**

Разработать программу, где реализуется поиск того или иного значения с клавиатуры

**Лабораторное задание:****Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа  $G$ . Выведите матрицу на экран.
2. Для сгенерированного графа осуществите процедуру поиска расстояний, реализованную в соответствии с приведенным выше описанием. При реализации алгоритма в качестве очереди используйте класс `queue` из стандартной библиотеки C++.
3. Реализуйте процедуру поиска расстояний для графа, представленного списками смежности.

**Задание 2**

1. Реализуйте процедуру поиска расстояний на основе обхода в глубину.
2. Реализуйте процедуру поиска расстояний на основе обхода в глубину для графа, представленного списками смежности.
3. Оцените время работы реализаций алгоритмов поиска расстояний на основе обхода в глубину и обхода в ширину для графов разных порядков.

**Листинг:**

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
```

```

#include <queue>

using namespace std;

void BFSD(int** G, int numG, int* dist, int s){
    queue<int> q;
    int v;

    dist[s] = 0;
    q.push(s);

    while (!q.empty()) {
        v = q.front();
        q.pop();
        printf("%3d",v);

        for (int i = 0; i < numG; i++){
            if (G[v][i] == 1 && dist[i] == -1){
                q.push(i);
                dist[i] = dist[v]+1;
            }
        }
    }

    printf("\nDistance from %d to:\n", s);
    for(int i=0;i<numG; i++){
        printf("%d : %d\n", i, dist[i]);
    }
}

int main() {
    int** G;
    int numG, current;
    int* dist;

    srand(time(NULL));

```

```

printf("Input number of vertission: ");
scanf("%d", &numG);

dist = (int*)malloc(numG * sizeof(int));
G = (int**)malloc(numG * sizeof(int*));
for (int i = 0; i < numG; i++){
    G[i] = (int*)malloc(numG * sizeof(int));
}

for (int i = 0; i < numG; i++){
    dist[i] = -1;
    for (int j = i; j < numG; j++){
        G[i][j] = G[j][i] = (i==j ? 0 : rand() % 2);
    }
}

for (int i = 0; i < numG; i++){
    for (int j= 0; j< numG; j++){
        printf("%3d", G[i][j]);
    }
    printf("\n");
}

printf("Input start vert: ");
scanf("%d", &current);

printf("Path:");
BFSD(G, numG, dist, current);

for (int i = 0; i < numG; i++)
    free(G[i]);
free(G);
free(dist);

 getch();

```

```
    return 0;  
  
}
```

### **Выводы:**

В ходе выполнения лабораторной работы были разработаны программы, выполняющие указанные в лабораторной работе задачи. Результаты работ программ совпали с результатами трассировок, следовательно, программы работают без ошибок.

Получили опыт в создании проектов в среде MicrosoftVisualStudio.