# Санкт-Петербургский государственный политехнический университет

Кафедра “Прикладная математика”

Отчет по лабораторной работе 3

“Алгоритмы и структуры данных”

Вариант 2

Студент группы № 5030102/20003

ФИО Крехов Данил Андреевич

Выполнил 15.11.2023

**Оглавление**

[**Постановка задачи** 3](#_gjdgxs)

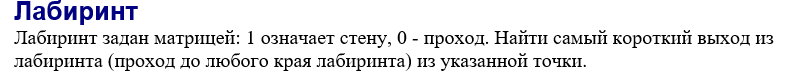
[**Описание алгоритма** 3](#_30j0zll)

[**Текст программы** 5](#_1fob9te)

[**Описание тестирования** 11](#_3znysh7)

# **Постановка задачи**

Задача этой лабораторной работы заключается в осуществлении алгоритма с возвратом. Условие моего варианта следующее:



# **Описание алгоритма**

Пройду по порядку выполнения программы:

Если текущая клетка = целевая, то завершить поиск с успехом (true).  
Для очередной клетки повернуть в каждое доступное направление, кроме того, откуда пришли в эту клетку, т.е.:  
пойти налево, если это возможно И пришли не справа. Если путь успешен – завершить поиск;  
пойти направо, если это возможно И пришли не слева. Если путь успешен – завершить поиск;  
пойти вверх, если это возможно И пришли не снизу. Если путь успешен – завершить поиск;  
пойти вниз, если это возможно И пришли не сверху. Если путь успешен – завершить поиск;  
Завершить поиск с неуспехом .

# **Текст программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#define \_USE\_MATH\_DEFINES

#include <stdio.h>

#include <stdlib.h>

#include <stdbool.h>

#include <math.h>

#define INT\_MAX 2147483647

//строит лабиринт в виде двумерного массива, прочитанного с файла(n - ширина, m - длина лабиринта)

int\*\* CreateLabirint(char\* filename, int n, int m) {

//lab\_t lab;

FILE\* file = fopen(filename, "r");

if (file == NULL) {printf("file error");exit(11);}

int\*\* A = (int\*\*)malloc(sizeof(int\*) \* n);

if (A == NULL) {printf("memory error");exit(21);}

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

A[i] = (int\*)malloc(sizeof(int) \* m);

if (A[i] == NULL) {printf("memoery error"); exit(211);}

}

for (int i = 0; i < n && !feof(file) && !ferror(file); i++)

{

for (int j = 0; j < m && !feof(file) && !ferror(file); j++)

{

A[i][j] = 0;

fscanf(file, "%d ", &A[i][j]); printf("%d ", A[i][j]);

}

printf("\n");

}

return A;

}

//рекурсивная функция, находящая самый короткий путь от заданных координат до символа '9' в лабиринте

//a - лабиринт, row,col -координаты начала, min\_l -минимальная длина , cur\_l -текущая длина,visit- вспомогательный массив со следом

void dir(int\*\* a, int row, int col, int \*min\_l,int cur\_l, int\*\*visit) {

if (a[row][col] == 9) {

\*min\_l = min(cur\_l, \*min\_l);

// if(min\_l>2){printf("%i \n", min\_l);}

}

//меняем на 1 поле, где мы прошли

visit[row][col] = 1;

//проверяем на свободную ячейку в соновном массиве и всподручном

if (a[row + 1][col] != 1 && visit[row+1][col] !=1) {

cur\_l++;

dir(a, row + 1, col, min\_l,cur\_l, visit);

}

if (a[row][col + 1] != 1 && visit[row ][col+1] != 1) {

cur\_l++;

dir(a, row, col + 1, min\_l, cur\_l, visit);

}

if (a[row - 1][col] != 1 && visit[row-1][col] != 1) {

cur\_l++;

dir(a, row - 1, col, min\_l, cur\_l , visit );

}

if (a[row][col - 1] != 1 && visit[row ][col-1] != 1) {

cur\_l++;

dir(a, row, col - 1, min\_l, cur\_l , visit);

}

//очищаем поле, чтобы найти другой путь

visit[row][col] = 0;

}

//обработка функции

int Shortlength(int\*\* a, int i, int j ,int n,int m) {

int\*\* tmp = (int\*\*)malloc(sizeof(int\*) \* n);

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

tmp[i] = (int\*)malloc(sizeof(int) \* m);

}

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

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

tmp[i][j] = 0;

}

}

int min\_dist = INT\_MAX;

dir(a, i, j, &min\_dist, 0, tmp);

if (min\_dist != INT\_MAX) {

return min\_dist;

}

}

//длина и ширина(n и m задаются вручную)

int main() {

int n = 9;

int m = n;

//tmp - вспомогательный массив

printf("laba 3 variant 2 labirint Krehov danil 5030102\_20003\n\n");

int\*\* A = CreateLabirint("C:/Users/danil/Desktop/labirint.txt", n, m);

printf("\n\n");

printf("\n");

int b = 9999999;

int R = Shortlength(A, 1, 1,n,m);

printf("%i ", R);

printf("\n");

return 0;

}

# **Описание тестирования**

Приведу скриншоты вывода программы

Для всех тестов начинал с координаты (1,1)

Входные данные:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Вывод программы:

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Входные данные:

Входные данные (решение отсутствует):

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Вывод программы:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание