

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет Информатика и вычислительная техника

Кафедра Кибербезопасность информационных систем

**ПРАКТИЧЕСКАЯ РАБОТА № 4**

на тему «Графы и алгоритмы на графах – 1: хранение графа в памяти, DFS, BFS, топологическая сортировка, Дейкстра, Форд-Беллман, Флойд»

Выполнила обучающаяся гр. ВКБ32

Дедиченко Дарья Валентиновна

Проверил:

Доцент, Савельев Василий Александрович

Ростов-на-Дону

2021

А. Два коня

#include <fstream>

#include <vector>

#include <queue>

#include <algorithm>

#include <string>

using namespace std;

const int INF = 1000000000;

int di[8] = { -2, -2, -1, 1, 2, 2, -1, 1 };

int dj[8] = { -1, 1, -2, -2, -1, 1, 2, 2 };

int get\_num(char letter) {

    if (letter == 'a') {

        return 1;

    }

    else if (letter == 'b') {

        return 2;

    }

    else if (letter == 'c') {

        return 3;

    }

    else if (letter == 'd') {

        return 4;

    }

    else if (letter == 'e') {

        return 5;

    }

    else if (letter == 'f') {

        return 6;

    }

    else if (letter == 'g') {

        return 7;

    }

    else if (letter == 'h') {

        return 8;

    }

}

vector<vector<int>>d1;

void bfs1(int ui, int uj) {

    queue<pair<int, int>>q;

    q.push({ ui, uj });

    d1[ui][uj] = 0;

    while (!q.empty()) {

        ui = q.front().first;

        uj = q.front().second;

        q.pop();

        for (int h = 0; h < 8; h++) {

            int toi = ui + di[h];

            int toj = uj + dj[h];

            if (toi < 8 && toi > -1 && toj > -1 && toj < 8) {

                if (d1[toi][toj] == INF) {

                    d1[toi][toj] = d1[ui][uj] + 1;

                    q.push({ toi, toj });

                }

            }

        }

    }

}

vector<vector<int>>d2;

void bfs2(int ui, int uj) {

    queue<pair<int, int>>q;

    q.push({ ui, uj });

    d2[ui][uj] = 0;

    while (!q.empty()) {

        ui = q.front().first;

        uj = q.front().second;

        q.pop();

        for (int h = 0; h < 8; h++) {

            int toi = ui + di[h];

            int toj = uj + dj[h];

            if (toi < 8 && toi > -1 && toj > -1 && toj < 8) {

                if (d2[toi][toj] == INF) {

                    d2[toi][toj] = d2[ui][uj] + 1;

                    q.push({ toi, toj });

                }

            }

        }

    }

}

int main() {

    ifstream fin("input.txt");

    ofstream fout("output.txt");

    string first, second;

    fin >> first >> second;

    int x1 = get\_num(first[0]);

    int y1 = first[1] - '0';

    int x2 = get\_num(second[0]);

    int y2 = second[1] - '0';

    //fout << x1 << " " << y1 << endl << x2 << " " << y2;

    d1.resize(8, vector<int>(8, INF));

    bfs1(x1 - 1, y1 - 1);

    d2.resize(8, vector<int>(8, INF));

    bfs2(x2 - 1, y2 - 1);

    int miny = INF;

    for (int i = 0; i < 8; i++) {

        for (int j = 0; j < 8; j++) {

            if (d2[i][j] != 0 && d1[i][j] != 0 && d1[i][j] + d2[i][j] < miny) {

                miny = d1[i][j] + d2[i][j];

            }

        }

    }

    if (miny == INF) {

        fout << -1;

    }

    else {

        fout << miny / 2;

    }

}

