中山大学计算机学院

本科生课程报告

（2024 学年秋季学期）

课程名称：数据结构与算法

班级	6 班	专业（方向）	计算机科学与技术（人工智能与大数据）
学号	23336217	姓名	覃司翰

一、 问题描述

设计一个校园导游程序，为来访的客人提供各种信息查询服务。

二、 编程环境

1. 编程语言

C++

2. 开发工具

Vscode

三、 设计思路

1. 算法描述

核心是用弗洛伊德算法在预输入阶段求出最短路径

2. 关键代码展示（带注释）

搜索部分

```
void print_place_info(string start, Info* H){
    ofstream outfile("C:/Users/20148/Desktop/Investigator/Tests/result.txt", ios::app);
    // check
    if (!outfile.is_open()) {
        cerr << "Failed to open file \"result.txt\"" << endl;
        return;
    }
    // output Log

    outfile << "Log: searching for info of " << start << "..." << endl;

    Info* search = H->next;
    bool found_flag = false;

    // search
```

```
        while (search!=nullptr) {
            if (start == search->code || start == search->name) {
                // output result
                outfile << search->info << endl;
                found_flag = true;
                break;
            }
            search = search->next;
        }

        // con't find
        if (!found_flag) {
            outfile << "could not find info of " << start << endl;
        }
}
```

最短路径（弗洛伊德算法）

```
void floydWarshall(vector<vector<int>>& dist, vector<vector<int>>& next, int n) {
        for (int i=0; i<n; i++) {
            for (int j=0; j<n; j++) {
                if (dist[i][j] != INT_MAX) {
                    next[i][j] = j;
                }
            }
        }

        for (int k=0; k<n; k++) {
            for (int i=0; i<n; i++) {
                for (int j=0; j<n; j++) {
                    if (dist[i][k] != INT_MAX && dist[k][j] != INT_MAX && dist[i][j] > dist[i][k]
+ dist[k][j]) {
                        dist[i][j] = dist[i][k] + dist[k][j];
                        next[i][j] = next[i][k];
                    }
                }
            }
        }
}
```

输出时调用回溯

```
void printpath(string start, string end, vector<vector<int>>& next){
        int start_index = start[0] - 'A';
        int end_index = end[0] - 'A';
```

```
stack<int> path;
int current = start_index;

ofstream outfile("C:/Users/20148/Desktop/Investigator/Tests/result.txt", ios::app);
outfile << "Log: searching for path from " << start << " to " << end << endl;

// fail
if (next[start_index][end_index] == -1) {
    if (outfile.is_open()) {
        outfile << "CANNOT find path from " << start << " to " << end << endl;
    }
    else {
        cerr << "Failed to open file \"result.txt\"" << endl;
    }
    return;
}

// exist
while (current != end_index) {
    path.push(current);
    current = next[current][end_index];
}
path.push(end_index);

//  倒置
stack<int> path_stack;
while (!path.empty()) {
    path_stack.push(path.top());
    path.pop();
}

// print
if (outfile.is_open()) {
    outfile << "Path from " << start << " to " << end << ": ";
    while (!path_stack.empty()) {
        char enroute = path_stack.top() + 'A';
        outfile << enroute << " ";
        path_stack.pop();
    }
    outfile << endl;
}
else {
    cerr << "Failed to open file \"result.txt\"" << endl;
}
```

```
        outfile.close();
    }
```

3. 创新点&优化（如果有）

使用弗洛伊德算法，在预输入阶段就完成最短路径计算，可以避免每次输入都调用一次最短路径计算，在面对大量数据的时候节省了计算资源

四、 结果及分析

1. 实验结果展示示例（可图可表可文字，尽量可视化）

输入 visitor.txt

```
info
    A
    find_path
    H A
    info
    B
    find_path
    K B
    info
    E
    find_path
    I E
    info
    H
    find_path
    G H
    info
    I
    find_path
    J I
    info
    K
    find_path
    A K
```

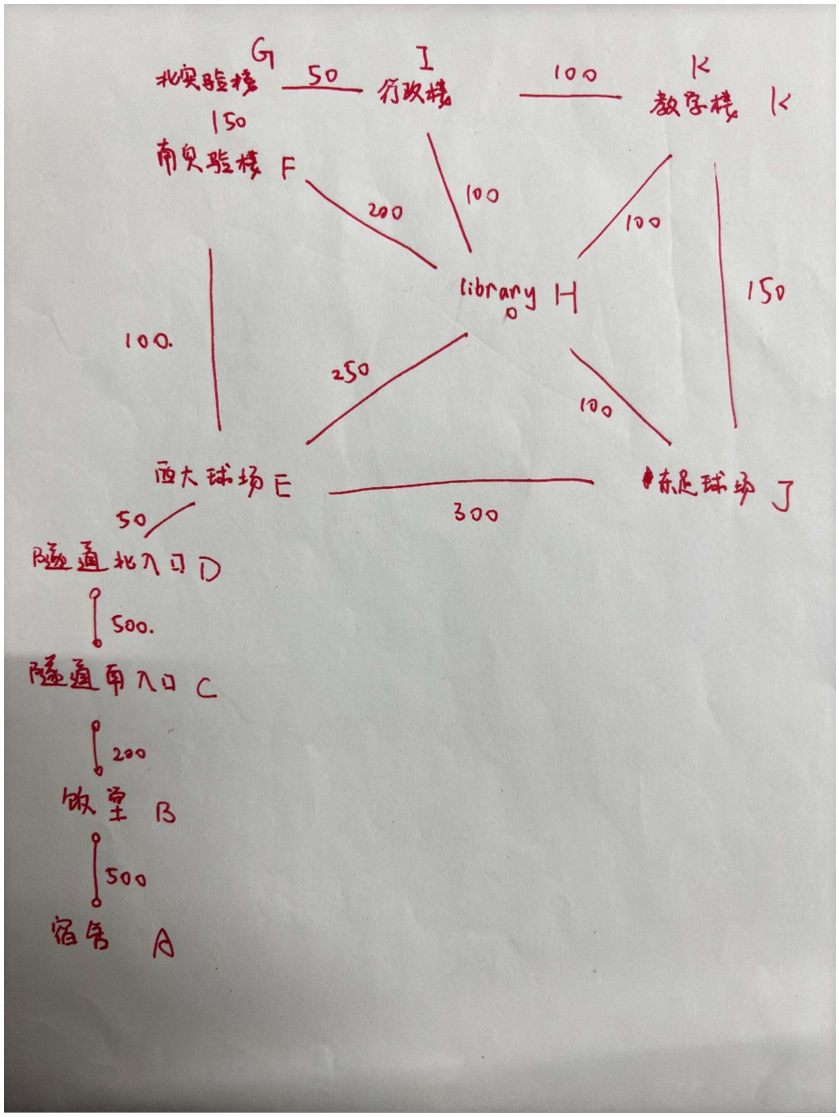
输出 result.txt

```
Log: searching for info of A...
沾湿昏暗草堂
Log: searching for path from H to A
Path from H to A: H E D C B A
Log: searching for info of B...
猪糠禽肴食槽
Log: searching for path from K to B
```

Path from K to B: K I G F E D C B
Log: searching for info of E...
深夜跑堂
Log: searching for path from I to E
Path from I to E: I G F E
Log: searching for info of H...
家
Log: searching for path from G to H
Path from G to H: G F H
Log: searching for info of I...
枢密院
Log: searching for path from J to I
Path from J to I: J H I
Log: searching for info of K...
昏睡牢房
Log: searching for path from A to K
Path from A to K: A B C D E F G I K

2. 评测指标展示及分析（可分析运行时间等）

符合预期，图如下：



五、 心得感想

这次比前两次大作业熟练很多，但是时间压力还是比较大的
在一开始我打算用迪杰斯特拉，做到一半突然想到这样可能需要调用很多次路径函数，最终决定用弗洛伊德来节省开销