**算法作业1**

**学号17263829 姓名赵宜珺**

**过桥问题**

**在一个月黑风高的夜晚，有N个人要过桥，他们只有一盏灯，桥很窄，每次最多只能允许两个人过桥，现在告诉你每个人过桥需要的时间，问你最少需要多少时间能使得所有的人都过桥。**

**(如果两个人的过桥时间不同，则他们一起过桥的时间按照过桥慢的人的时间算。)**

数据：1 3 6 8 12(单位:分钟) 需要时间为29分钟

1 2 5 10 (单位:分钟) 需要时间为17分钟

**算法思路**：

|  |
| --- |
| 先讲过河的人按过河时间排序。假设f[n]为n个人过河总共要用的时间，t[i]为第i个人过河需要的时间，则有:  f[1] = t[1],  f[2] = t[2],  f[3] = t[2]+t[1]+t[3],  f[4] = t[2]+t[1]+t[4]+t[2]+f(2)  f[n] = t[2]+t[1]+t[n]+t[2]+f[n-2], n >= 4 |

**程序代码：**

|  |
| --- |
| **#include<bits/stdc++.h>**  **using namespace std;**  **const int MAXN = 100;**  **int t[MAXN], N;**  **int dp(int n){**  **if(n <= 2){**  **return t[n]; //只有1或2个人**  **}**  **if(n == 3){ //有三个人**  **return t[1] + t[2] + t[3];**  **}**  **//有四个人或以上**  **int fn\_2 = t[2]; //n-2 = 2, n = 4**  **int fn\_1 = t[1] + t[2] + t[3]; // n-1 = 3, n = 4**  **int fn = 0;**  **for(int i = 4; i <= n; ++i){**  **fn = t[1] + 2\*t[2] + t[i] + fn\_2;**  **fn\_2 = fn\_1;**  **fn\_1 = fn;**  **}**  **return fn;**  **}**  **int main(){**  **cin >> N;**  **t[0] = 0;**  **for(int i = 1; i <= N; i++){**  **cin >> t[i];**  **}**  **sort(t, t + N);**  **cout << dp(N) << endl;**  **}** |

**运行截图：**

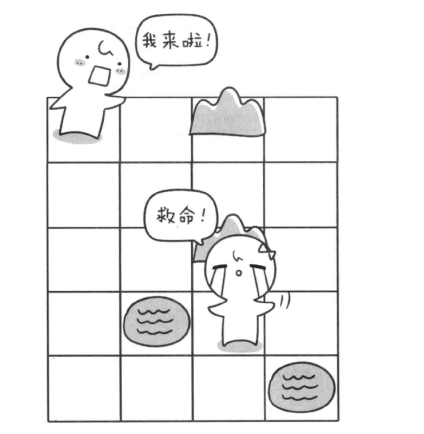
|  |
| --- |
|  |

**算法作业2**

**学号17263829 姓名赵宜珺**

**Maze 迷宫问题**

**请问从求援者从起点到施救者终点的位置，需要多少步？Steps**



//测试数据

5 4

0 0 1 0

0 0 0 0

0 0 1 0

0 1 0 0

0 0 0 1

1 1 4 3

// 1 1 表示主人出发的起点 4 3 表示被困者的坐标

输出：steps=7 需要7步才能解救被困者。

**算法思路**：

|  |
| --- |
| 迷宫用二维数组保存。  用一个向量表示行走的方向。  利用深度优先搜索尝试所有走法，找到步数最小的走法，打印输出 |

**程序代码：**

|  |
| --- |
| **#include <bits/stdc++.h>**  **using namespace std;**  **const int MAXN = 101;//迷宫最大大小**  **int maze[MAXN][MAXN] = {0};//迷宫**  **int route[MAXN][MAXN] = {0};//行走记录**  **int direct[4][2] = {{0, 1}, {1, 0}, {0, -1}, {-1, 0}};//顺时针方向**  **int n, m;//迷宫实际行和列**  **int sx, sy, ex, ey;//起点，终点**  **int minSteps = 0x7FFFFFFF;**  **void dfs(int x, int y, int step){**  **int cx, cy, i;**  **if(x == ex && y == ey){ //找到了**  **if(step < minSteps){**  **minSteps = step;**  **}**  **return;**  **}**  **for(i = 0; i <= 3; i++){ //四个方向尝试**  **cx = x + direct[i][0];**  **cy = y + direct[i][1];**  **if(cx < 1 || cx > n || cy < 1 || cy >m){**  **continue; //越界，继续尝试**  **}**  **if(maze[cx][cy] == 0 && route[cx][cy] == 0){**  **route[cx][cy] = 1;**  **dfs(cx, cy, step + 1);//走一步**  **route[cx][cy] = 0;**  **}**  **}**  **//尝试完**  **return;**  **}**  **int main() {**  **freopen("maze.in", "r", stdin);**  **cin >> n >> m;**  **for(int i = 1; i <= n; i++)**  **for(int j = 1; j <= m; j++)**  **cin >> maze[i][j];**  **cin >> sx >> sy >> ex >> ey;**  **route[sx][sy] = 1;**  **dfs(sx,sy,0);**  **cout << "minSteps= " << minSteps << endl;**  **}** |

**maze.in**

|  |
| --- |
| **5 4**  **0 0 1 0**  **0 0 0 0**  **0 0 1 0**  **0 1 0 0**  **0 0 0 1**  **1 1 4 3** |

**运行截图：**

|  |
| --- |
|  |