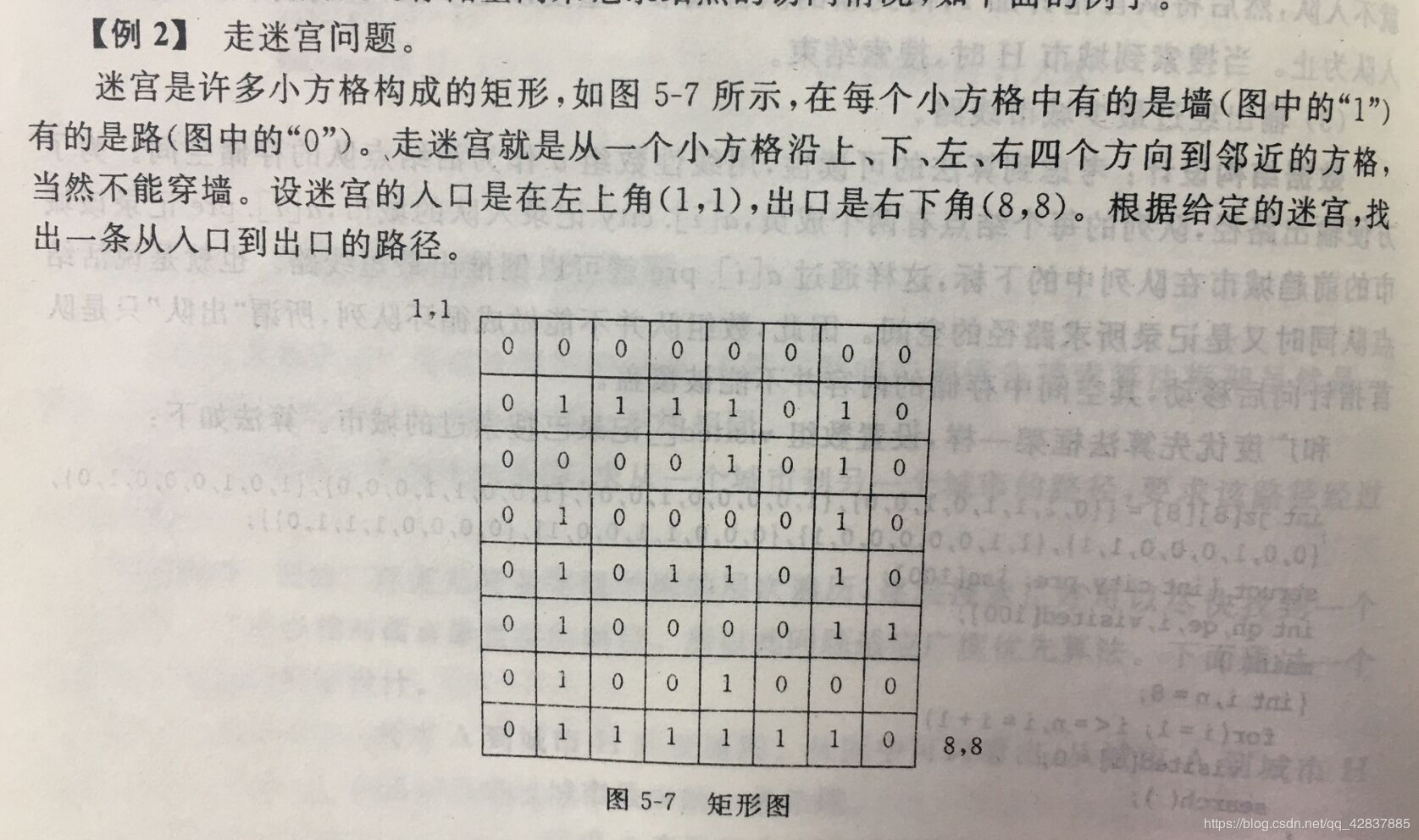
**走迷宫问题**



代码：

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include <stdio.h>

#include<stdlib.h>

#define LEFT 8

#define RIGHT 8

int maze[LEFT][RIGHT] = { { 0, 0, 0, 0, 0, 0, 0, 0 },

{ 0, 1, 1, 1, 1, 0, 1, 0 },

{ 0, 0, 0, 0, 1, 0, 1, 0 },

{ 0, 1, 0, 0, 0, 0, 1, 0 },

{ 0, 1, 0, 1, 1, 0, 1, 0 },

{ 0, 1, 0, 0, 0, 0, 1, 1 },

{ 0, 1, 0, 0, 1, 0, 0, 0 },

{ 0, 1, 1, 1, 1, 1, 1, 0 } };

struct

{

int x;

int y;

int pre;

}sp[100];

int fx[] = { 1, -1, 0, 0 };

int fy[] = { 0, 0, 1, -1 };

int ph, pe, i, j, k;

void out()

{

printf("(%d,%d)", sp[ph].x + 1, sp[ph].y + 1);

while (sp[ph].pre)

{

ph = sp[ph].pre;

printf("--(%d,%d)", sp[ph].x + 1, sp[ph].y + 1);

}

printf("\n");

}

int check(int i, int j)

{

if (i < 0 || i >= LEFT || j < 0 || j >= RIGHT)

return 0;

if (maze[i][j] == 1)

return 0;

return 1;

}

void search()

{

ph = 1;

pe = 0;//父亲节点

sp[0].x = 0;

sp[0].y = 0;

sp[0].pre = 0;

maze[0][0] = 1;

while (1)

{

pe++;

for (k = 0; k < 4; k++)

{

i = sp[pe].x + fx[k];

j = sp[pe].y + fy[k];

if (check(i, j) == 1)

{

ph++;

sp[ph].x = i;

sp[ph].y = j;

sp[ph].pre = pe;

maze[i][j] = 1;

if (sp[ph].x == LEFT - 1 && sp[ph].y == RIGHT - 1)

{

out();

return;

}

}

}

}

}

int main()

{

search();

system("pause");

return 0;

}

结果截屏

