# 棋盘填充

## 应承轩 201785071 电计1701

## 算法代码

#include <iostream>

using namespace std;

#define BoardSize 8

struct pt {

int x;

int y;

};

char board[BoardSize][BoardSize];

int tile = 0;

void ptBoard() {

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

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

cout << board[i][j] << ' ';

}

cout << endl;

}

cout << endl;

}

void initBoard() {

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

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

board[i][j] = '.';

}

}

}

void setPoint(struct pt t, char c) {

board[t.y][t.x] = c;

}

bool inSquare(struct pt o, struct pt b, int size) {

return b.x >= o.x && b.x < o.x + size &&

b.y >= o.y && b.y < o.y + size;

}

void solve(struct pt o, struct pt strange, int size) {

if (size == 1) return;

cout << "next step:" << endl;

ptBoard();

int s = size / 2;

int t = ++tile;

struct pt \_pt = o;

struct pt new\_st;

if (inSquare(\_pt, strange, s)) {

solve(\_pt, strange, s);

}

else {

new\_st = { o.x + s - 1, o.y + s - 1 };

setPoint(new\_st, 'A' + t);

solve(\_pt, new\_st, s);

}

\_pt = { o.x + s, o.y };

if (inSquare(\_pt, strange, s)) {

solve(\_pt, strange, s);

}

else {

new\_st = { o.x + s, o.y + s - 1 };

setPoint(new\_st, 'A' + t);

solve(\_pt, new\_st, s);

}

\_pt = { o.x, o.y + s };

if (inSquare(\_pt, strange, s)) {

solve(\_pt, strange, s);

}

else {

new\_st = { o.x + s - 1, o.y + s };

setPoint(new\_st, 'A' + t);

solve(\_pt, new\_st, s);

}

\_pt = { o.x + s, o.y + s };

if (inSquare(\_pt, strange, s)) {

solve(\_pt, strange, s);

}

else {

new\_st = { o.x + s, o.y + s };

setPoint(new\_st, 'A' + t);

solve(\_pt, new\_st, s);

}

}

void main() {

initBoard();

struct pt strange = { 6, 3 };

setPoint(strange, '@');

cout << "the init board:" << endl;

ptBoard();

solve({ 0, 0 }, strange, BoardSize);

ptBoard();

}

## 运行结果

**the init board:**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**. . . . . . . .**

**. C . . . . . .**

**. . . . . . . .**

**. . . B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D . . . . . .**

**D C C . . . . .**

**. . . . . . . .**

**. . . B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E . . . .**

**D C C E . . . .**

**. C . . . . . .**

**. . . B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E . . . .**

**D C C E . . . .**

**F C . . . . . .**

**F F . B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E . . . .**

**D C C E . . . .**

**F C G G . . . .**

**F F G B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E . . . .**

**D C C E . H . .**

**F C G G . . . .**

**F F G B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E I I . .**

**D C C E I H H .**

**F C G G . . . .**

**F F G B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G . H . .**

**F F G B . . @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H . .**

**F F G B K K @ .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**. . . B . . . .**

**. . . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**. . . B . . . .**

**. M . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N . B . . . .**

**N M . . . . . .**

**. . . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N O B . . . .**

**N M O O . . . .**

**. M . . . . . .**

**. . . . . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N O B . . . .**

**N M O O . . . .**

**P M M . . . . .**

**P P . . . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N O B B . . .**

**N M O O . . . .**

**P M M Q . . . .**

**P P Q Q . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N O B B . . .**

**N M O O . . . .**

**P M M Q . . . .**

**P P Q Q . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N O B B S . .**

**N M O O S S R .**

**P M M Q . . . .**

**P P Q Q . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N O B B S T T**

**N M O O S S R T**

**P M M Q . R . .**

**P P Q Q . . . .**

**next step:**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N O B B S T T**

**N M O O S S R T**

**P M M Q U R R .**

**P P Q Q U U . .**

**Final**

**D D E E I I J J**

**D C C E I H H J**

**F C G G K H L L**

**F F G B K K @ L**

**N N O B B S T T**

**N M O O S S R T**

**P M M Q U R R V**

**P P Q Q U U V V**