자료구조\_과제5\_01

Equivalence Relation

본 과제는 C++을 기반으로 작성하였습니다.



자료구조

김성열교수님

소프트웨어학과

201811182

박원준

2019.04.16

목 차

1. 문제 정의…………………………………………………………………………..p1

2. 소스코드…………………………………………………………………………….p3

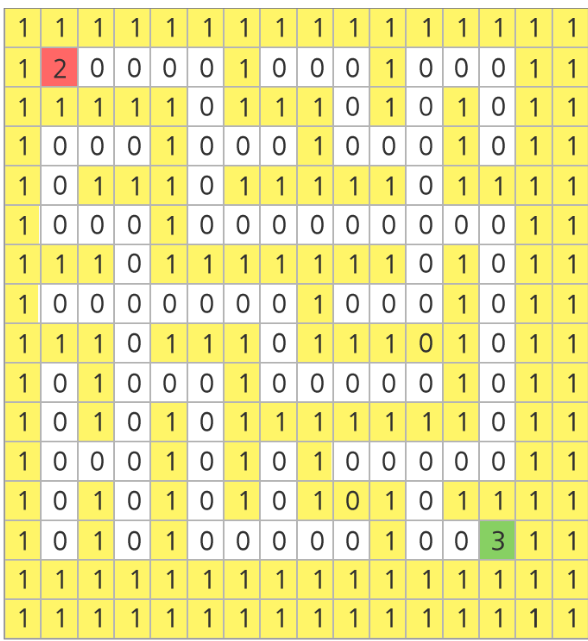
3. 실행 결과…………………………………………………………………………..p5

4. 참고문헌…………………………………………………………………………….p5

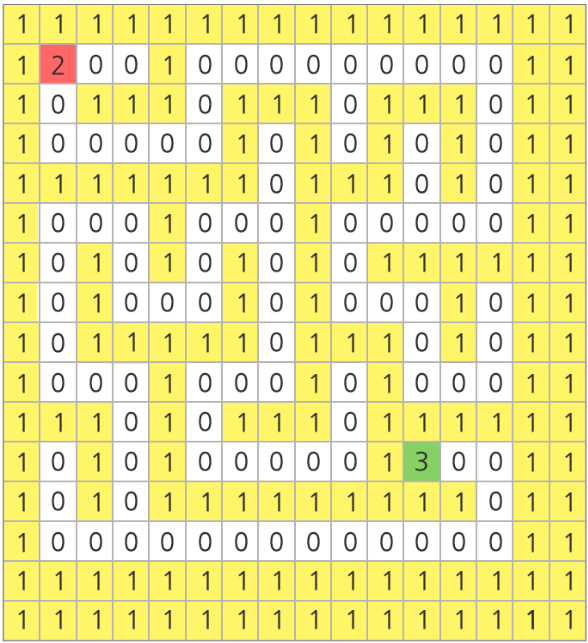
**1. 문제 정의**

아래 그림과 같은 미로가 있다. 16\*16 행렬의 형태로 만들어진 미로에서 흰색 바탕은 길, 노란색 바탕은 벽을 나타낸다. 가장 최상단에 있는 칸을 (0, 0)의 기준으로 하여, 가로방향을 x 방향, 세로방향을 y 방향이라고 할 때, 미로의 시작점은 (1, 1)이고 도착점은 (13, 13)이다. 주어진 미로의 출발점으로부터 도착지점까지 갈 수 있는 길이 있는지 판단하는 프로그램을 작성하라.

아래의 예시에서는 도달 가능하다.



아래의 예시에서는 출발점이 (1, 1)이고, 도착점이 (11, 11)이며 도달이 불가능하다.



[입력]

각 테스트 케이스의 첫 번째 줄에는 테스트 케이스의 번호가 주어지며, 바로 다음 줄에 테스트 케이스가 주어진다. 총 10개의 테스트 케이스가 주어진다.

테스트 케이스에서 1은 벽을 나타내며 0은 길, 2는 출발점, 3은 도착점을 나타낸다.

[출력]

#부호와 함께 테스트 케이스의 번호를 출력하고, 공백 문자 후 도달 가능 여부를 1 또는 0으로 표시한다. (1 – 가능함, 0 – 가능하지 않음)

**2. 소스코드**

/\*

\* 미로찾기

\* Quest URL : https://www.swexpertacademy.com/main/code/problem/problemDetail.do?contestProbId=AV14vXUqAGMCFAYD&categoryId=AV14vXUqAGMCFAYD&categoryType=CODE&&&

\* 박원준

\*/

#include <stack>

#include <vector>

#include <iostream>

using namespace std;

vector<int> escape;

bool FindExit(const int& idx, char maze[10][16][16])

{

bool exit = false;

stack<pair<int, int>> pos;

// 시작점 index는 [1][1].

// 0 : 갈 수 있는곳 1 : 갈 수 없는 곳 2 : 시작점 3 : 도착점

pos.push(pair<int, int>(1, 1)); // 시작점

while (!pos.empty())

{

int i = pos.top().first;

int j = pos.top().second;

maze[idx][i][j] = '1'; pos.pop(); // index [i][j]는 이미 방문한 곳. 이제 방문할 수 없는 곳으로 만들어 줌.

// 왼쪽

switch (maze[idx][i][j - 1])

{

case '0':

pos.push(pair<int, int>(i, j - 1));

break;

case '1': break;

case '3':

while (!pos.empty())

pos.pop();

exit = true;

break;

}

// 오른쪽

switch (maze[idx][i][j + 1])

{

case '0':

pos.push(pair<int, int>(i, j + 1));

break;

case '1': break;

case '3':

while (!pos.empty())

pos.pop();

exit = true;

break;

}

// 위

switch (maze[idx][i - 1][j])

{

case '0':

pos.push(pair<int, int>(i - 1, j));

break;

case '1': break;

case '3':

while (!pos.empty())

pos.pop();

exit = true;

break;

}

// 아래

switch (maze[idx][i + 1][j])

{

case '0':

pos.push(pair<int, int>(i + 1, j));

break;

case '1': break;

case '3':

while (!pos.empty())

pos.pop();

exit = true;

break;

}

}

return exit;

}

int main(void)

{

char maze[10][16][16];

for (int i = 0; i < 10; i++)

{

int tmp; cin >> tmp; // TestCase Number;

for (int j = 0; j < 16; j++)

for (int k = 0; k < 16; k++)

cin >> maze[i][j][k];

}

for (int i = 0; i < 10; i++)

FindExit(i, maze) ? escape.push\_back(1) : escape.push\_back(0);

for (int i = 0; i < escape.size(); i++)

cout << "#" << i + 1 << " " << escape[i] << endl;

return 0;

}

**3. 실행 결과**

TestCase는 <SW Expert Academy 1226. [S/W 문제해결 기본] 7일차 – 미로1>에서 제공하는 기본 input값이다. 그에 따른 실행 결과는 다음과 같다.

 22:33:35 실행이 완료되었습니다. 실행 시간 : 0.00836s

 22:33:35 #1 1  
#2 1  
#3 1  
#4 0  
#5 1  
#6 1  
#7 0  
#8 1  
#9 1  
#10 1

 22:33:35 실행을 시작합니다.

 22:33:35 성공적으로 컴파일 되었습니다.

 22:33:35 Input data가 없으므로 Sample input으로 테스트됩니다.

**4. 참고 문헌**

미로찾기 문제 : SW Expert Academy 1226. [S/W 문제해결 기본] 7일차 – 미로1