

자료구조응용

08. 미로찾기 (5점)

2022.3.30.

1. [정적할당 배열을 이용한 미로탐색] Program 3.12 Maze search function을 사용하는 미로탐색 프로그램을 작성하여라.(5점)

[프로그램 설명]

다음 ① ~ ⑤의 변수는 모두 전역으로 선언되며 정적할당을 사용함.

① maze

- 입구는 left top, 출구는 right bottom으로 가정
- 미로 데이터입력 ("[input.txt](#)", 최대 10×10 행렬, 첫 줄은 미로의 행과 열의 크기)

4	5			
0	0	0	1	1
1	1	0	0	0
1	0	1	1	1
1	0	1	0	0

entrance (1, 1), exit (4, 5)

② mark

③ move

```
typedef struct {
    short int vert;
    short int horiz;
} offsets;
offsets move[8];
```

Name	Dir	move[dir].vert	move[dir].horiz
N	0	-1	0
NE	1	-1	1
E	2	0	1
SE	3	1	1
S	4	1	0
SW	5	1	-1
W	6	0	-1
NW	7	-1	-1

※ move 선언 시 배열 초기화 목록을 사용하자!

④ stack

```
#define MAX_STACK_SIZE 100
typedef struct {
    short int row;
    short int col;
    short int dir;
} element;
element stack[MAX_STACK_SIZE];
int top = -1;
```

※자료구조응용 06. 2번의 스택 소스를 활용할 것

⑤ 기호상수 : #define TRUE 1
 #define FALSE 0

전역변수 : int EXIT_ROW, EXIT_COL;

```

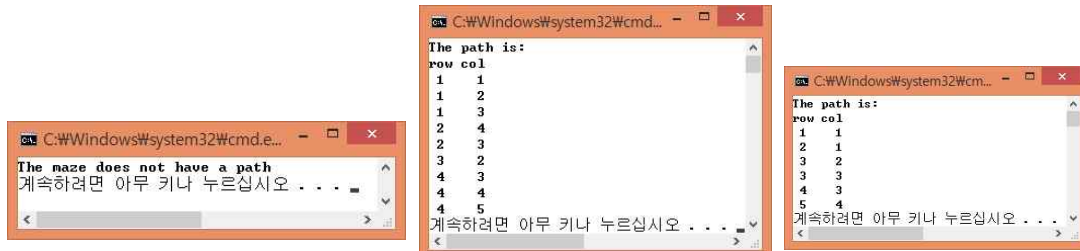
void path(void)
{
    /* output a path through the maze if such a path exists */
    int i, row, col, nextRow, nextCol, dir, found = FALSE;
    element position;
    mark[1][1] = 1; top = 0;
    stack[0].row = 1; stack[0].col = 1; stack[0].dir = 1;
    while (top > -1 && !found) {
        position = pop();
        row = position.row; col = position.col;
        dir = position.dir;
        while (dir < 8 && !found) {
            /* move in direction dir */
            nextRow = row + move[dir].vert;
            nextCol = col + move[dir].horiz;
            if (nextRow == EXIT_ROW && nextCol == EXIT_COL)
                found = TRUE;
            else if (!maze[nextRow][nextCol] &&
                !mark[nextRow][nextCol]) {
                mark[nextRow][nextCol] = 1;
                position.row = nextRow; position.col = nextCol;
                position.dir = ++dir;
                push(position);
                row = nextRow; col = nextCol; dir = 0;
            }
            else ++dir;
        }
    }
    if (found) {
        printf("The path is:\n");
        printf("row col\n");
        for (i = 0; i <= top; i++)
            printf("%2d%5d", stack[i].row, stack[i].col);
        printf("%2d%5d\n", row, col);
        printf("%2d%5d\n", EXIT_ROW, EXIT_COL);
    }
    else printf("The maze does not have a path\n");
}

```

Program 3.12: Maze search function

[실행예]

4 5	4 5	5 4
0 0 0 1 1	0 0 0 1 1	0 1 1 1
1 1 0 0 0	1 1 0 0 0	0 1 1 0
1 0 1 1 1	1 0 1 1 1	1 0 0 1
1 0 1 0 0	1 0 0 0 0	1 1 0 1
		1 0 1 0



■ 제출 형식

- 솔루션 이름 : DS 08
- 프로젝트 이름 : 1
- 각 소스파일에 주석처리
“학번 이름”

“본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.”

- 실행화면을 캡처한 보고서를 작성 후 pdf 파일로 변환하여 솔루션 폴더에 포함
- 솔루션 정리 메뉴를 수행 후 전체 솔루션을 “학번.zip”으로 압축하여 제출

■ 주의

- 소스 복사로는 실력향상을 기대할 수 없습니다!!!
- 1차 마감 (LMS 과제 마감일) : 수업일 자정
- 2차 마감 (LMS 과제 열람기한) : 수업 익일 자정(만점의 50%, 반올림)
- 1, 2차 마감 이외의 제출은 허용하지 않습니다. (이메일 제출 불가!)