과제 II - 미로 찾기(Stack, Queue)

1 과제 개요

스택과 큐를 각각 활용하여 시작점 s 에서 도착점 d 까지의 경로를 찾는 프로그램을 작성한다. 스택과 큐로 "갈 곳" 후보를 관리해야 하며, 최종 경로를 저장한다. 스택을 써서 찾을 때는 "아무 경로나" 찾으면 되고, 큐를 쓸 때는 최단 경로를 찾아야 한다.

2 미로 정보

미로는 한변의 길이가 9 인 정사각형 모양이며 이차원 배열로 주어진다. 0 은 지나갈 수 있는 통로를 의미하고, 1 은 지나갈 수 없는 벽을 의미한다. 주어진 미로의 가장자리는 모두 벽으로 되어 있다. 출발점과 도착점의 정보는 행과 열 숫자로 주어지며, 출발점과 도착점의 행과 열 범위는 1 부터 7 까지이고, 항상 미로에서 지나갈 수 있는 통로에 있다. 또한 출발점부터 도착점까지 도달하는 길이 반드시 존재한다고 가정한다. 길의 이동 방향은 상하좌우만 가능하고, 대각선으로는 이동할 수 없다.

3 기능 요구 사항

void find_maze_stack(int** maze, int start_row, int start_col, int end_row, int end_col) 함수와 void find_maze_queue(int** maze, int start_row, int start_col, int end_row, int end_col) 함수를 구현한다. 두 함수는 갈 곳 후보를 스택과 큐로 관리한다는 점이 다르다.

매개변수 maze 가 미로 정보가 있는 이차원 배열이고, start_row 가 시작점의 행 번호, start_col 이 시작점의 열 번호, end_row 가 도착점의 행 번호, end_col 이 도착점의 열 번호다.

두 함수 모두 실행하고 나면 maze 이차원 배열에 출발점부터 도착점까지의 경로에 숫자 2 를 저장한다.

4 테스트 함수

미로찾기 함수를 테스트를 하기 위해 다음 코드를 쓸 수 있다.

```
/* Print the maze (len:9) to the terminal */
void print_maze_9(int** maze)
{
        int nrows = 9;
        int ncols = 9;

        /* print * for wall, blank for open space, o for path */
        for (int row = 0; row < nrows; row++) {
                for (int col = 0; col < ncols; col++) {
                        if (maze[row][col] == 1)
                                printf("*");
                        else if (maze[row][col] == 0)
                                printf(" ");
                        else if (maze[row][col] == 2)
                                printf("o");
                        else
                                printf(" ", maze[row][col]);
                }
                printf("\n");
        }
}

/* Test the maze solving algorithm */
void maze_test_9x9()
{
        /* define a simple maze (9x9) */
        int maze1[9][9] = {
                {1, 1, 1, 1, 1, 1, 1, 1, 1},
                {1, 0, 0, 0, 1, 0, 0, 0, 1},
                {1, 0, 1, 0, 1, 0, 1, 1, 1},
                {1, 0, 1, 0, 0, 0, 0, 1, 1},
                {1, 0, 1, 1, 1, 1, 1, 1, 1},
                {1, 0, 0, 0, 0, 0, 0, 1, 1},
                {1, 1, 1, 1, 1, 1, 0, 1, 1},
                {1, 0, 0, 0, 0, 0, 0, 0, 1},
                {1, 1, 1, 1, 1, 1, 1, 1, 1}
        };

        /* convert maze to array of pointers for compatibility */
        int *maze_row_ptrs[9];
```

```
        for(int i = 0; i < 9; i++)
                maze_row_ptrs[i] = maze1[i];

        /* print the maze */
        printf("Maze:\n");
        print_maze_9(maze_row_ptrs);
        printf("\n\n");

        int start_row = 1, start_col = 1;
        int end_row = 7, end_col = 7;

        /* solve the maze using stack-based approach */
        printf("Solving maze using stack-based approach:\n");

        find_maze_stack(maze_row_ptrs, start_row, start_col, end_row, end_col);

        print_maze_9(maze_row_ptrs);
        printf("\n\n");

        /* reset the maze */
        int maze2[9][9] = {
                {1, 1, 1, 1, 1, 1, 1, 1, 1},
                {1, 0, 0, 0, 1, 0, 0, 0, 1},
                {1, 0, 1, 0, 1, 0, 1, 1, 1},
                {1, 0, 1, 0, 0, 0, 0, 1, 1},
                {1, 0, 1, 1, 1, 1, 1, 1, 1},
                {1, 0, 0, 0, 0, 0, 0, 1, 1},
                {1, 1, 1, 1, 1, 1, 0, 1, 1},
                {1, 0, 0, 0, 0, 0, 0, 0, 1},
                {1, 1, 1, 1, 1, 1, 1, 1, 1}
        };
        for(int i = 0; i < 9; i++)
                maze_row_ptrs[i] = maze2[i];

        /* solve the maze using queue-based approach */
        printf("Solving maze using queue-based approach:\n");

        find_maze_queue(maze_row_ptrs, start_row, start_col, end_row, end_col);

        print_maze_9(maze_row_ptrs);
        printf("\n\n");

        return;
}
```

5 테스트 결과

미로찾기 함수를 완성한 후 테스트를 하면 다음과 같이 화면에 출력된다.

```
Maze:
*********
*   *   *
* * * ***
* *    **
* *******
*       **
****** **
*        *
*********


Solving maze using stack-based approach:
*********
*o*******
*o*******
*o*******
*o*******
*oooooo**
******o**
*    *oo*
*********


Solving maze using queue-based approach:
*********
*o  *   *
*o* * ***
*o*    **
*o*******
*oooooo**
******o**
*     oo*
*********
```

6 제출물

코드 + 보고서를 함께 제출한다 (설계 결정, 올바름 근거, 세부 코드 테스트 전략 간단히 기술)

제출 요령: 과제 1 과 동일하게 서면 또는 이메일 제출

제출 기한: 10 월 14 일 화요일 수업시간까지