<자료구조>

Programming 연습문제 1



|  |  |
| --- | --- |
| 학부 | 컴퓨터공학과 |
| 학번 | 20171327 |
| 이름 | 조민아 |

1. 교재 3.5절 연습문제 5번 (maze에서 path 찾기 프로그램 완성)의 답안이 교재의 홈페이지 (https://www.cise.ufl.edu/~sahni/fdsc2ed/)의 Exercise solutions Chap 3의 8번 항 (page 127, #5)에 제시되어 있다.

(a) 제시된 프로그램은 교재에서 공부한 프로그램과 어떤 차이점이 있는지 설명하시오.

1) 교재에서 공부한 프로그램은 방향에 대한 인덱스가 북쪽부터 0으로 시작하지만, 제시된 프로그램에서는 방향에 대한 인덱스가 북쪽부터 1로 시작한다.

그러므로 방향의 조건이 들어가는 반복문에서는 조건에서 약간의 변화가 있다.

예를 들어 교재에서 공부한 프로그램에서는 while문에서 dir<8 && !found라고 조건이 제시 되어있지만, 제시된 프로그램에서는 while문에서 dir<=8 && !found라고 조건이 제시되어 있다.

2) 또한 교재에서 공부한 프로그램은 미로를 통과하는 경로를 출력할 때에 방향을 출력하지 않지만, 제시된 프로그램에서는 방향도 함께 출력한다.

(b) 제시된 프로그램에서 발견한 path를 출력할 때 이동 방향을 출력하고 있는데 단, 마지막 이동인 EXIT으로의 이동 방향은 출력하지 않는다. 이를 출력하려면 제시된 프로그램을 어떻게 수정해야 하는지 설명하시오.

if (next\_row == NUM\_ROWS && next\_col == NUM\_COLS) {

/\* path has been found, exit loop and print it out \*/

position.row = row; position.col = col;

position.dir = ++dir;

add(position);

found = TRUE;

}

제시된 프로그램에서 마지막 이동 방향은 스택에 담지 않기 때문에 방향을 출력할 수 없다.

그러므로 다음으로 이동할 수 있는 경로가 출구 경로일 때, 스택에 현재 행과 열에 대한 위치를 넣어주고, 다음으로 탐색할 수 있는 방향의 경로를 스택에 넣어준 후, 나중에 경로를 출력할 때에 스택에서 행과 열은 그대로 출력하고, 방향의 인덱스는 1을 뺀 후에 출력해주면 된다.

2. eclass에는 또다른 버전의 maze 프로그램이 제시되어 있다. 이 프로그램은 교재에서 공부한 내용에 서 아래 수정 사항을 반영하여 작성한 것이다.

(1) 출구(EXIT) 대신 임의의 위치(row, col)를 목적지(destination)로 지정하 여 입구에서 목적지까지의 path를 찾는다.

(2) mark[][] 배열을 사용하지 않는다.

(3) 어떤 위치로 처음 갔을 때 8 방향 이웃 위치 중에 목적지가 있으면 path 찾기가 종료되도록 한다.

(4) path 찾기를 위해 이동할 방향의 결정은 매번 random하게 한다..

(a) eclass에 제시된 프로그램에서 path() 함수의 빈 곳 ‘가’, ‘나’에 들어갈 C 코드를 쓰시오.

- ‘가’

row = position.row;

col = position.col;

dir\_vector = position.dir\_vector;

dir = get\_next\_dir(init\_dir\_vector(row,col));

* ‘나’

if (nextRow == DST\_ROW && nextCol == DST\_COL)

{

found = TRUE;

}

else {

maze[nextRow][nextCol] = -1;

position.row = row; position.col = col;

position.dir\_vector = get\_next\_dir(init\_dir\_vector(row, col));

push(position);

row = nextRow; col = nextCol;

dir\_vector = init\_dir\_vector(row, col);

dir = get\_next\_dir(dir\_vector);

}

(b) ‘가’, ‘나’에 작성한 C 코드에 대해 왜 그렇게 작성하였는지 이유, 불필요한 연산을 수행한 것은 없는 지, 기타 설명이 필요한 부분에 대해 설명하시오.

‘가’

먼저 행과 열에 방향 벡터에 스택에 들어있는 값으로 초기화해준 후에, 현재 위치에서 갈 수 있는 모든 방향을 init\_dir\_vector 함수를 이용해 계산해주고, 이를 토대로 갈 수 있는 방향 중에서 한 방향을 랜덤으로 설정하여 이동할 경로를 설정하기 위해 get\_next\_dir 함수를 사용한다.

‘나’

‘가’ 과정 후에 경로를 이동시켜야 하는데, 만약 이동할 경로가 마지막 경로이면 found 변수에 참값을 넣어 while문을 빠져나올 수 있게 해주고, 만약 이동할 경로가 마지막 경로가 아니라면, 이동할 경로는 이미 지나간 경로가 될 것 이므로, 그 위치를 -1로 지정해준다. 그 후 스택에 현재 위치를 넣어주고, 위에서 이동할 경로를 -1로 지정해줬으므로, 이를 토대로 현재 위치에서 이동할 수 있는 경로를 설정해주고, 이를 토대로 이동할 경로를 랜덤으로 설정하여 스택에 넣어준다. 그 후에, 현재 위치를 이동할 위치로 옮겨준 후에, 다시 이동할 수 있는 경로를 설정한 후, 갈 수 있는 방향을 랜덤으로 정한 후에 while문을 다시 돈다.

(c) (a)에서 완성한 프로그램을 실행한 화면, 사용한 10 by 10 maze 데이터 및 목적지 좌표를 제시하시오. 설명이 필요한 부분이 있으면 설명하시오

* 사용한 데이터

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

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

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

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

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

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

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

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

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

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

* 목적지

#define DST\_ROW 5 //목적지 좌표 row

#define DST\_COL 6 //목적지 좌표 col

* 실행화면

