실험 PRJ-2 미로(Maze) 1주차 예비보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20201635 이름: 전찬

**1. 목적**

미로 생성 알고리즘에는 어떤 알고리즘이 있는지 알아본다. 또한 Eller’s algorithm 으로 완전 미로를 제작할 때 필요한 자료구조를 설계한다. 설계한 자료구조에 대한 시간 및 공간 복잡도를 보인다.

**2. 다른 미로 생성 알고리즘**

강의자료에서 설명되어 있는 Eller’s의 알고리즘은 한 줄씩 미로를 만들어내기 때문에 NXM의 미로에서 O(M\*N)의 형태로 빠른 시간 내에 미로를 만들어내는 장점이 존재한다. 하지만 그 효율성에 비해 직관적으로 이해하기 어렵다는 단점이 존재한다.

이에 비해, 재귀적 백트래킹 알고리즘(Recursive Backtracking Algorithm)은 직관적으로 이해하기 쉬운 방법으로 미로를 만들어낼 수 있다. 재귀적 백트래킹 알고리즘은 일정한 시작점부터 랜덤하게 주변의 위치로 이동하며 미로를 제작해내는 방법인데, 지나갔던 경로를 stack에 push하며, 막다른 길에 도달했을 때(주변이 모두 이미 거쳐간 위치일 때), 이동이 가능한 위치까지 pop을 해주며 미로를 만들어내는 알고리즘이다. 이렇게 계속 push와 pop을 통해서 이동하며 미로를 생성하고, 미로의 모든 위치에 방문을 완료했을 때, 종료하는 알고리즘이다. 하지만 위에서 말한 것처럼, 시간 복잡도와 공간 복잡도가 Eller’s Algorithm에 비해서 좋지는 않다. stack을 사용하며, 직접 방문하는 형태이기 때문이다.

**3. 미로를 제작할 때 필요한 자료구조와 시간 및 공간 복잡도**

개인적으로 생각한 것은, 두 가지 형태인데, 하나의 방법은 각 칸의 좌표와 각 칸의 값, 그 칸의 오른쪽 bar, 아래 bar(벽면)을 하나의 구조체로 정의해서 수행하는 방식이 있을 것 같다. 또 다른 방식으로는 각 칸의 값을 하나의 좌표로 표현하고 배열로 표현하며, 아래 벽면과 오른쪽 벽면에 대해서 또 다른 배열들로 표현하는 방법이 있다.

이 중에서 역시 구조체로 정의해서 수행하는 방법이 미로 제작 알고리즘을 만들어낼 때 적합한 것 같다.

결국 위의 구조체를 사용하면, 필요한 함수는 현재 줄을 초기화하는 함수(각 칸의 값이 정해지지 않았을 때, 새로운 값을 집어넣는 역할을 수행한다.), 현재 줄의 각 column마다 구조체를 랜덤으로 합쳐주는 함수(왼쪽 구조체를 기준으로 오른쪽 bar을 부셔준다.), 또한 랜덤으로 아래로 한 칸 내려주는 함수(같은 숫자를 기준으로 아래로 한 칸 내려간 것이 하나도 없다면, 다시 반복한다.) 가 필요하다. 위의 함수들을 계속해서 반복해줌을 통해서 완벽한 미로를 만들어낼 수 있다.

위 함수들의 시공간 복잡도를 알아보면, 아래와 같다.(공간 복잡도에서, 이미 구조체의 모든 원소를 미리 정의해 놓는다고 가정해서, 모두 공간 복잡도가O(1) 형태이다.)

-현재 줄을 초기화해주는 함수

시간 복잡도: 한 줄을 읽어가며 merge를 수행하기 때문에, 시간 복잡도는 O(column)

공간 복잡도: O(1)

-column마다 구조체를 랜덤으로 합쳐주는 함수

시간 복잡도: 한 줄을 읽어가며 merge를 수행하기 때문에, 시간 복잡도는 O(column)

공간 복잡도: O(1)

-랜덤으로 아래로 한 칸 내려주는 함수

시간 복잡도: 현재 줄에서 한 줄을 읽어가며, 아래로 내려주거나 내려주지 않기 때문에, 시간 복잡도는 O(column)

공간 복잡도: O(1)