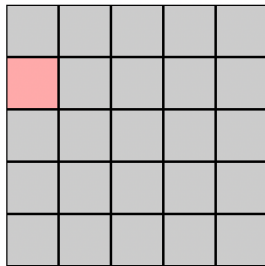


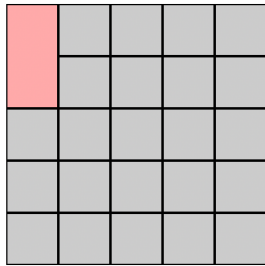
1. Eller's algorithm 외 알고리즘을 조사하고 그 방법을 기술하시오.

<Recursive Backtracking>

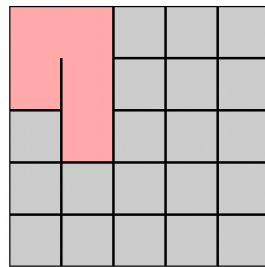
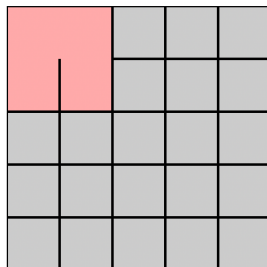
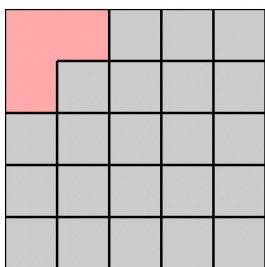
M*N의 미로 맵에서 시작점을 하나 정한다. 네 가지 방향 중 깎을 수 있는 방향에 대해 어떤 벽을 깎을지 하나 랜덤하게 고른다. 그 방향으로 이웃하는 칸을 아직 방문하지 않았다면 벽을 깎고 그 칸으로 이동한다. 만약 이웃하는 칸 중 갈 수 있는 칸이 없다면 이전 칸으로 돌아간다. 돌아가서 또 랜덤하게 벽을 선택해 벽을 깎고 이동하는 것을 반복하여 시작점까지 돌아가게 된다면 미로 생성을 종료한다.



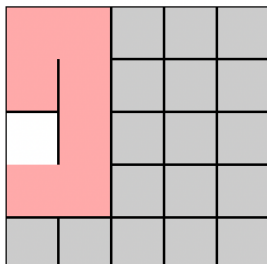
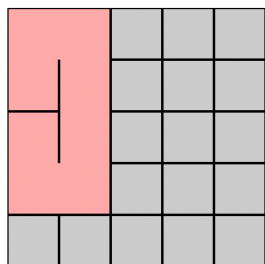
랜덤하게 시작점을 선택한다.



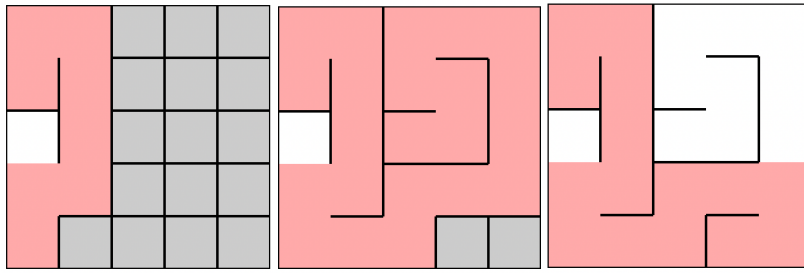
랜덤하게 한 방향에 대해 벽을 깎고 이동한다.



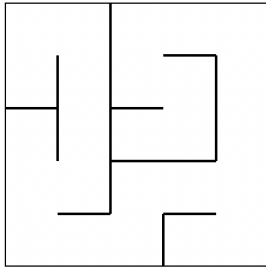
계속 벽을 깎으며 이동한다.



더 이상 벽을 깎고 이동할 곳이 없으면, 돌아간다.



곳이 없으면 돌아가는 작업을 반복한다.



이 알고리즘은 방법이 직관적이라는 장점이 있지만 메모리 공간을 많이 사용한다는 단점이 있다.

2. 완전 미로(Perfect Maze)를 만들기 위해 선택한 알고리즘 구현에 필요한 자료구조를 설계하고 기술하시오. 설계한 자료구조를 사용하였을 경우 선택한 알고리즘의 시간 및 공간복잡도를 보이시오.

세로 벽과 가로 벽을 저장할 배열이 필요하다. 그리고 해당 칸이 어느 집합에 속하는지 저장할 배열이 필요하다. 이 배열들의 크기는 각각 $O(\text{Maze의 가로 길이(WIDTH)})$ 이다. 따라서 Eller's algorithm을 사용해 완전 미로를 만들 경우 공간복잡도는 $O(\text{WIDTH})$ 이다.

한편 미로의 각 행을 두 번 이상 보지 않으며 각 행은 WIDTH개의 칸으로 이루어져있고, 행이 HEIGHT만큼 있기 때문에 시간 복잡도는 $O(WIDTH * HEIGHT)$ 이다.