

1. 실험시간에 작성한 프로그램의 알고리즘과 자료구조를 요약하여 기술하시오. 완성한 알고리즘의 시간 및 공간 복잡도를 보이고 실험 전에 생각한 방법과 어떻게 다른지 아울러 기술하시오.

### 1) 알고리즘

Eller's Algorithm을 사용하였다. 미로의 첫번째 줄 부터 차례대로 훑으며 바로 오른쪽 칸 사이의 벽을 제거할 것인지를 임의로 선택하고(첫번째 단계), 첫번째 줄을 모두 훑은 후에 다시 한번 훑으며(두번째 단계) 바로 아래 칸 사이의 벽을 제거할 것인지를 임의로 선택한다. 첫번째 단계에서 벽을 제거하는 경우 두개의 칸은 같은 번호의 집합에 속하게 되며, 두번째 단계에서 벽을 제거하는 경우 줄 하나에 존재하는 있는 집합마다 최소 한개의 칸은 바로 아래칸과의 벽이 제거되어야 한다.

### 2) 자료구조

2차원 구조체 배열 자료구조를 사용하였다. 각 구조체 변수는 집합 번호와 위, 아래, 왼쪽, 오른쪽 벽 정보를 저장하는 필드를 가진다. 집합 번호는 말 그대로 집합 번호 값을 가지며, 벽 정보는 값이 1이면 벽이 존재하고, 0이면 벽이 존재하지 않음을 의미한다. 미로에는 총  $N \times M$ 개의 칸이 존재하므로 2차원 구조체 배열 역시  $N \times M$ 개의 구조체 변수를 가진다.

### 3) 시간 및 공간 복잡도

시간복잡도와 공간복잡도는  $O(N \times M)$ 으로 동일하다.  $N$ 과  $M$ 은 각각 미로의 가로 길이와 세로 길이이며, 총  $N \times M$ 개의 미로 칸이 존재한다. 모든 칸을 저장하기 위해  $O(N \times M)$ 의 공간복잡도가 필요하며 집합 번호가 바뀔 때 마다 미로의 전체 칸을 훑으면서 찾아 바꿔줘야 하기 때문에  $O(N \times M)$ 의 시간복잡도를 가지게 된다.

### 4) 실험 전에 생각한 방법과 다른 점

바로 다음 칸과 바로 아랫줄의 칸만 고려하면 된다고 생각했는데, 예를들어 바로 다음 칸 또는 아랫줄의 칸의 집합 번호가 4에서 1로 바뀌는 경우 현재 미로에 존재하는 모든 4번 칸의 집합 번호를 1로 변경시켜줘야 된다는 사실을 코드를 작성하면서 깨달았다. 때문에 실험 전에 생각했던 것 보다 좀 더 복잡한 코드를 작성하게 되었다.