**학과: 컴퓨터공학과 학년: 2 학번: 20191657 이름:최세은**

**# 미로(maze) 1주차**

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

우선 미로를 만들기 위해서 자료구조로 동적할당 된 배열을 사용하였다. 그리고 요구한대로 eller의 maze 알고리즘을 사용하였다.

미로를 만들기 위해 input으로 받은 n \* 2 + 1을 높이로, m \* 2 + 1을 너비로 하는 int형 배열을 생성해 메모리를 동적으로 할당했다. 그리고 +, -, | 기호들을 이용해 격자무늬로 만들어준다. 따라서 배열은 n개의 열과 m개의 행으로 이루어진 nm개의 방을 갖고, 각 방들은 사면이 벽으로 막혀있다. 그리고 eller의 알고리즘을 이용해 알맞게 벽을 뚫어가며 미로를 만든다. 실습에서는 요구한대로 완전 미로를 만들었다. 우선 미로의 첫번째 줄, maze[1]을 roomnum 변수를 이용해 서로 다른 집합에 속하도록 초기화한다. 그리고 for 반복문을 이용해 위에서부터 차례대로 한 줄씩 미로를 만들어간다. 현재 행에서 첫 번째 방부터 마지막 방까지 반복문으로 차례대로 비교하며 인접한 두 방이 서로 다른 집합에 속해있을 때 rand()%2 난수를 이용해 사이의 벽을 제거할지 안할지 임의로 선택한다. 만약 벽을 제거하면 두 방이 연결되므로 같은 집합으로 합쳐지게 한다. 그리고 다음 줄로 수직 경로를 만드는데, 이때 각 집합마다 하나 이상을 임의로 선택하게 하기 위해 flag를 이용한다. 만약 다음 집합을 넘어갔는데도 수직 경로가 만들어지지 않아 flag가 0이라면 이전 집합의 마지막 방에서 수직 경로를 만든다. 그리고 수직 경로가 만들어지지 않은 방은 새로운 집합으로 초기화해준다. 마지막 줄에 도달하면 반복문을 빠져나와, 새로운 반복문을 통해 인접해 있으면서 다른 집합에 속하는 방들 사이의 모든 벽을 제거한다.

위 알고리즘의 시간 복잡도는 이중 반복문을 이용하므로 O(n \* m), 공간 복잡도는 이중 배열을 사용하였으므로 O(n \* m)이다.

실험 전에 조사했었던 크루스칼 알고리즘이랑 비교하면, 위 eller 알고리즘은 위에서부터 한 줄씩 차례대로 알고리즘을 수행하며 크루스칼 알고리즘은 랜덤으로 인접한 두 방을 선택해 시작하며 계속 인접한 두 방을 랜덤으로 선택해 비교한다.

1. **숙제문제를 해결하기 위한 알고리즘 및 자료구조를 요약하여 기술하시오. 시간 및 공간 복잡도를 보이시오.**

이번주에 과제는 존재하지 않는다.

**# WaterFall 2주차**

1. **물이 흐르는 것을 표현하기 위해서, 본인이 구현한 알고리즘과 자료구조를 기술한다.**

우선 물이 흐르는 과정을 선분을 이용해서 화면에 표현하였다. 따라서 알고리즘으로는 경로 계산 알고리즘, 자료구조로는 구조체 배열을 사용했다.

‘s’를 누르면 물이 흐르기 시작한다. 따라서 fall\_flag를 1로 설정해준다. 그리고 물이 흐르는 과정을 점들을 연결해서 표시할 것이기 때문에 DNODE 구조체 집합 water\_arr을 사용했다. 우선 for 반복문으로 input으로 미리 받았던 선분들의 x 좌표들을 비교하여 현재 좌표(water\_x, water\_y)가 그 사이에 존재할 때 해당하는 선분들의 index를 배열 x에 저장해둔다. 이때 만약 현재 좌표 밑으로 아무런 선분이 존재하지 않을 경우 water\_y를 728로 설정하여 바로 맨 아래에 도달하게 한다. 그리고 해당하는 선분들 중 현재 좌표보다는 아래면서 가장 위에 존재해서 물이 떨어질 때 닿을 선분을 찾는다. 이는 직선의 방정식을 이용해서 찾았는데, 물이 떨어질 때 x 좌표는 그대로인 점을 이용해서 선분들의 직선의 방정식을 구한 후 x 좌표를 대입해 y 좌표를 구했다. 그리고 y 좌표들을 비교해서 가장 작은 좌표를 가지는 선분에 물이 떨어질 예정이므로, 해당 y좌표를 water\_y에 저장하고 water\_arr에 현재 좌표를 추가한다. 그 다음 선분에 물이 떨어졌으므로 물이 선분을 따라 선분의 끝까지 흐르는 것을 구현한다. 해당 선분의 양 끝점의 y좌표를 비교하여, 더 큰 y좌표의 점으로 물이 떨어지므로 water\_arr에 해당사항을 반영해준다. 이 과정을 반복문을 이용해서 물이 바닥에 닿을 때까지, 즉 water\_y가 728이 될 때까지 반복해준다. 그리고 draw 함수에서 water\_arr에 저장된 점들을 연결한 선분들을 화면에 표시하여 물이 어떻게 흘렀는지 나타낸다.