11주차 결과보고서

전공: 아트&테크놀로지학과 학년: 4학년 학번: 20191048 이름: 김도솔

**1.**

알고리즘:

1. 미로의 가로와 세로를 결정하여 미로 배열을 할당한다.
2. 미로 배열을 초기화한다. 짝수 행과 짝수 열은 벽('+'), 짝수 행과 홀수 열은 수평 벽('-'), 홀수 행과 짝수 열은 수직 벽('|')로 설정한다. 홀수 행과 홀수 열은 미로의 경로를 나타내는 공백(' ')으로 초기화한다.
3. 각 방은 고유한 집합 번호를 가질 수 있도록 첫 번째 행을 초기화한다.
4. 두 번째 행부터 마지막 직전 행까지 다음 작업을 반복한다.
   * 현재 행에서 왼쪽 방과 오른쪽 방이 서로 다른 집합에 속해 있다면, 임의로 벽을 제거하고 두 방을 같은 집합으로 합친다.
   * 현재 행과 다음 행 사이의 벽을 설정한다. 오른쪽 방과 같은 집합에 속하는 경우, 임의로 벽을 제거한다.
   * 왼쪽 방과 같은 집합에 속하는 경우, 임의로 벽을 제거한다. 다음 행에서 벽을 제거하기 위해 플래그를 설정한다.
   * 왼쪽 방과 다른 집합에 속하는 경우, 벽을 제거한다.
   * 다음 행의 나머지 방들을 구체화한다.
5. 마지막 행의 벽을 설정한다.
6. 생성된 미로를 파일에 출력한다.

자료구조:

2차원 문자열 배열인 maze를 사용해 미로의 형태를 나타냈다. 짝수 행과 짝수 열은 벽('+'), 짝수 행과 홀수 열은 수평 벽('-'), 홀수 행과 짝수 열은 수직 벽('|')로 설정한다. 홀수 행과 홀수 열은 미로의 경로를 나타내는 공백(' ')으로 채운다.

**2.**

시간 복잡도:

미로 배열 초기화: 미로의 각 칸마다 초기화 작업이 필요하므로 O(n \* m) 시간이 소요된다.

첫 번째 행 설정: 미로의 가로 길이에 비례하는 O(n) 시간이 소요된다.

두 번째 행부터 마지막 직전 행까지 반복: 각 행에서 최대 O(n)번의 연산이 수행되므로, 전체 반복문은 O(m)번 실행된다. 따라서 이 단계에서는 O(n \* m) 시간이 소요된다.

마지막 행 설정: 미로의 가로 길이에 비례하는 O(n) 시간이 소요된다.

미로 출력: 미로의 모든 칸을 한 번씩 방문하므로 O(n \* m) 시간이 소요된다.

따라서 전체 알고리즘의 시간 복잡도는 O(n \* m)이다.

공간 복잡도:

미로 배열을 저장하기 위해 O(n \* m) 크기의 2차원 문자열 배열이 필요하기 때문에 알고리즘의 공간 복잡도는 O(n \* m)이다.

실험 전에는 3단계에서 랜덤 수직 경로 생성 시, 집합 내부에서 경로 생성 여부만 따지면 될 줄 알았는데, 실제로 코드를 구현해보니 방 위치, 집합 비교 등 생각보다 고려해야 할 경우의 수가 많아서 case 나누기가 까다로웠다. 또한 2단계, 6단계에서 set number만 비교해 다르면 벽을 제거하도록 알고리즘을 구성하면 되겠다고 생각했는데, 그렇게만 코드를 작성하면 순환미로가 생겨서 set number를 통일하기 위해 for 문을 추가하는 방식을 사용했다.