11주차 결과보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20211558 이름 : 윤준서

**1. 알고리즘과 자료구조**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Tree의 Union-Find 알고리즘을 이용하여 미로의 벽을 없앤다.

우선 본인의 부모 노드와 높이(레벨)의 정보를 저장한 구조체를 정의한다.

Find() 함수는 본인이 속한 구역이 몇인지 재귀적으로 함수를 호출하여 반환한다. 두 노드의 해당 값이 다른 경우 두 방은 서로 구분된 구역에 각자 속한 것이다.

Union() 함수는 두 구역을 비교해서 크기가 더 큰 구역을 기준으로 두 구역을 합병한다. 그 과정으로 level\_i = Find(uf, i)와 level\_j = Find(uf, j)를 이용해 두 구역의 크기를 저장하고, 둘을 비교한다. 두 구역의 크기가 같은 경우, 한 쪽의 크기를 1 늘리고 두 구역을 같은 방식으로 합병한다.

텍스트, 폰트, 대수학, 스크린샷이(가) 표시된 사진

자동 생성된 설명

구조체 Union-Find의 배열 형태 uf를 생성한다. 이 때 배열의 크기는 미로의 모든 칸 수인 n\*m으로 한다. parent는 각자 자신으로, level은 0으로 초기화한다.

while문을 이용할 때, 서로 같은 구역에 속한 방의 개수를 count에 저장한다. 배열의 크기인 n\*m과 같을 경우, 탐색이 모두 끝난 것이므로 break한다.

무작위로 칸을 지정하기 위해 각각 세로 좌표와 가로 좌표인 rand\_i, rand\_j를 정의하고, 오른쪽 혹은 아래쪽 방향으로 랜덤으로 진행할 방향 변수 direct를 정의한다.

조건문을 확인해본다. direct = 0인 경우 오른쪽, 1인 경우 아래로 칸을 이동한다. 칸의 인덱스 범위를 초과하는지 확인한다. 이후 선택한 칸과 이동한 칸의 구역이 서로 다른지 확인한다. 모든 것이 충족되면 Union()함수를 통해 두 구역을 합병한다.

**2. 예비보고서와 다른 점**

예비보고서에선 구조체의 이름을 union으로, 합병 대회의 함수명을 sum으로 쓴 것 이외에는 큰 차이점이 없다. 두 구역을 합병하기 위해 각 속한 구역의 레벨을 비교하여 상위와 하위 관계를 결정하는 과정을 추가했다.

위의 알고리즘의 시간 복잡도는 count를 통한 O(n \* m), Find과 Union을 통한 O(log(n \* m))이므로, 결론적으로 O(n \* m \* log(n \* m))이다. 공간 복잡도는 배열의 크기이므로 O(n \* m)이다.