11주차 결과보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20192135 이름: 윤영인

**1.**

엘러 알고리즘과 Union-Find 자료구조를 이용하여 랜덤으로 미로를 생성하였다.

1) 전역 변수

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

char \*\*maze: 미로를 그릴 2차원 배열 포인터

row, col: 미로의 실제 너비와 높이

N, M: 사용자로부터 입력 받은 너비와 높이

set: 각 칸이 속한 번호의 집합을 표현하기 위한 배열 포인터 변수

2) initParent()

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

각 칸에 번호를 설정하여 부모 노드를 지정하는 함수이다. 각 칸이 속한 번호의 집합을 표현하기 위해 set을 미로의 칸 수만큼 동적할당을 하여 순차적으로 칸에 0~(N\*M-1) 숫자를 저장한다. 해당 함수의 시간 복잡도 및 공간 복잡도는 O(N\*M) = O(사용자가 입력한 미로의 너비 \* 높이)이다.

3) findParent(int x)

텍스트, 폰트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

부모 노드를 찾는 함수로, 재귀적으로 부모 노드를 찾는다. 이 함수의 시간복잡도는 기본적으로 O(1)이지만, 부모 노드를 찾기 위한 경로가 길어질 경우, 즉 최악의 경우엔 O(log N)이다. 공간 복잡도는 set 배열만 사용하므로 O(1)이다.

4) unionParent()

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

두 노드의 부모를 합치는 함수이다. 인자로 받아온 x와 y중에 더 작은 수를 부모로 지정한다. 이 함수의 시간 복잡도는 O(1)이지만 3)에서 언급했듯이 findParent가 최악의 경우 O(log N)이므로 이 함수 역시 최악의 경우 O(log N)이다. 공간 복잡도는 set 배열만 사용하므로 O(1)이다.

5) MakeMaze()

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

랜덤으로 미로를 생성하는 함수이다. initParent 함수로 set 배열을 초기화하여 부모 노드를 지정한다. 미로를 탐색하면서 curNum을 설정하는데, 이 변수는 현재 위치인 (i, j)가 몇 번째 칸인지를 나타낸다. findParent로 curNum의 부모를 찾고, 현재 위치에서 한 칸의 너비인 2만큼 오른쪽으로 이동한 위치가 미로 안에 있다면 오른쪽에 벽이 있는 것이므로 오른쪽 벽을 뚫기 위해 다음 코드를 수행한다. 미로 안에 있다면, 현재 위치에서 한 칸 아래로 이동한 위치가 미로를 벗어났거나 혹은 벗어나지는 않았지만 랜덤 조건이 성립할 경우 오른쪽 칸 번호의 부모 노드를 찾는다. 이때 랜덤 조건은 rand() % 2가 0인지 아닌지를 판단하는데, 이는 벽을 뚫을지, 말지의 2가지 조건에 대한 조건을 판단하기 위해 2로 지정하였다. 만약 오른쪽 칸의 부모 노드와 현재 칸의 부모 노드가 같지 않다면 벽을 뚫을 수 있으므로 현재 위치에서 오른쪽 벽을 뚫고 부모 노드를 합친다.

다음으로 아래 벽을 뚫기 위해 현재 위치에서 아래로 한 칸 이동한 위치가 미로 안에 있는지를 판단한다. 안에 있다면, 아래쪽에 벽이 있는 것이므로 아래쪽 벽을 뚫기 위해 다음 코드를 수행한다. 현재 위치에서 오른쪽으로 한 칸 이동한 위치가 미로를 벗어나거나, 벗어나지는 않았지만 랜덤 조건이 성립할 경우 아래쪽 벽을 뚫는다. 현재 칸에서 한 칸 아래에 있는 칸의 번호는 M(col/2)만큼 차이가 난다. 따라서 현재 번호인 curNum보다 M만큼 큰 숫자의 부모 노드를 찾고 만약 현재 번호의 부모 노드와 같지 않다면 아래쪽 벽을 뚫고 두 노드의 부모를 합친다. 미로를 모두 탐색하여 벽을 뚫었다면 free(set)을 하여 동적할당을 해제한다. 이 함수의 시간 복잡도는 O(row\*col)이며, 공간 복잡도는 maze와 set 배열의 크기에 결정되므로 O(row\*col + N\*M)이다.

6) FreeMaze()

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

동적 할당한 2차원 배열 포인터인 maze를 할당 해제하는 함수이다. 시간 및 공간 복잡도는 O(row\*col)이다.

7) PrintMaze()

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

MakeMaze로 생성한 maze를 출력하는 함수이다. 이 함수의 시간 복잡도는 O(row\*col)이고 공간 복잡도는 O(1)이다.

8) DrawMaze()

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

MakeMaze로 생성한 maze를 .txt 파일에 그리는 함수이다. 이 함수의 시간 복잡도는 O(row\*col)이며 공간 복잡도는 O(1)이다.

9) main

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

사용자로부터 미로의 너비와 높이를 입력 받는다. 입력 받은 너비와 높이로 실제 미로에 사용될 너비와 높이를 계산하여 row와 col에 저장한다. row와 col만큼 2차원 maze 배열을 동적할당하고 rand() 함수를 사용하기 위해 srand(time(NULL))을 사용하여 랜덤으로 시드 값을 생성한다. InitMaze 함수로 칸이 채워진 미로를 그리고, MakeMaze로 랜덤으로 미로를 생성한다. PrintMaze와 DrawMaze로 터미널에 미로를 출력 및 텍스트 파일에 미로를 그린다. FreeMaze로 동적 할당한 maze 포인터 배열을 할당 해제한다.

**2. 예비 보고서와 달라진 점**

예비 보고서에선 프림 알고리즘을 사용하였다. 해당 알고리즘은 미로를 탐색하면서 방문했는지 확인하고 방문하지 않았으면 힙에 해당 칸을 넣고 pop한 칸과 인접한 칸들을 탐색하면서 벽을 뚫는다. 실습시간에 작성한 코드는 엘러 알고리즘을 사용하여 미로를 탐색하면서 칸이 서로 같은 집합에 속해 있는지 판단하고, 속하지 않았으면 벽을 뚫는 작업을 수행한다. 따라서 이번 실습 시간에 엘러 알고리즘으로 구현한 코드는 프림 알고리즘보다 간선의 수가 적고, 랜덤으로 미로를 생성하기 때문에 다양한 미로를 생성할 수 있다는 장점이 있다.