11주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211522 이름: 김정환

**1.**

.................

실습에서 작성한 프로그램은 Eller’s algorithm을 사용하였다. 또한 자료구조는 Simple array를 사용하여 구현하였다.

텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

void maze\_gene() 함수에 maze를 생성하는 기능을 모두 구현하도록 하였는데, 위의 코드는 해당 함수 내에서 maze의 첫 번째 줄을 생성하는 기능이다. 첫 번째 줄에서는 우선 외벽을 먼저 출력한 후, random하게 첫 세로 벽의 배치, 즉 ‘|’을 어디어디 출력하는지 추가하는 과정이다. 각각의 벽 위치가 생성함에 따라 set의 숫자들이 나눠지도록 해주고, 따로 구현한 vert\_print()로 vert 배열에 따라서 ‘|’와 ‘ ‘을 구분하여 출력하도록 하였다.

스크린샷, 텍스트, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

2번째부터 마지막 줄 전까지는 같은 과정이 반복된다. 따라서 이 부분에 대해서는 가로 줄을 출력해줄 hori\_line()과 세로 줄을 출력해줄 vert\_line()으로 함수를 생성했다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

void hori\_line()은 가로 줄, 즉 세로로 어떻게 길이 연결되는지를 정해 출력한다. 이는 한 줄을 모두 탐색하면서 hori array에 랜덤으로 0이나 1을 저장한다. 0인 경우가 길이다. 그리고 set당 하나 이상은 길이 있는지 확인 후 없으면 마지막에 하나 추가한다. 그 후 file에 해당 line을 출력한다.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

void vert\_line()은 N-1까지만 탐색하면서 바로 뒤의 set과 다른 set일 때만 랜덤으로 벽 여부를 결정한다. line\_flag는 1일 때 길이 되고 0일 때는 벽이 되는데 이는 vert에 나타내는 값과 반대이다. 또한 서로 벽이 없어져 합쳐질 때는 set의 숫자가 같아지도록 탐색하면서 설정한다. 그 후 vert\_print()로 해당 line을 출력한다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

maze\_gene()의 마지막 줄은 반복이 아니지만 가로 벽은 같은 방식이므로 hori\_line()으로 처리한다. 마지막 세로 줄은 vert\_line()과의 차이점으로는 서로 set이 다른 경우에는 랜덤이 아닌 무조건 0으로 합치는 방식이라는 점에서 차이가 있다. 그 후 vert\_print()로 출력해주고, 마지막 줄의 외벽도 출력하면서 종료한다. 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

main 함수에서는 전역 변수들로 생성한 N, M, fp, hori, vert, map에 대한 처리를 해준다. N, M은 maze의 가로, 세로이고, hori, vert는 각각 벽 여부를 저장할 배열, map은 한 줄씩 set의 숫자를 저장하는 배열이다. 그 후 maze\_gene()를 호출하여 maze를 maze.maz에 저장하고 메모리와 파일을 모두 해제하며 종료한다.

예비 보고서에서 작성하였을 때는 maze의 array를 전체 다 저장할 것으로 예상하였고, 전체 알고리즘은 모두 한 번씩 방문하므로 시간 복잡도가 O(N \* M), 공간 복잡도 역시 O(N\*M)이 될 것으로 예상하였다. 그러나 실제 구현에 있어서는 한 줄 씩만 저장하여 계산에 이용하여도 maze 출력에 아무런 지장이 없어 배열을 가로 길이인 N만큼의 배열 3개만을 이용하였다. 또한 탐색에 있어서 vert\_line()에서 서로 합치는 과정에서 set의 number를 같게 만드는 과정에서 가로 길이인 N만큼의 탐색이 한 번 더 있었다. 따라서 작성한 프로그램의 시간 복잡도는 O(N \* N \* M)이고, 공간 복잡도는 O(N)이다.

................