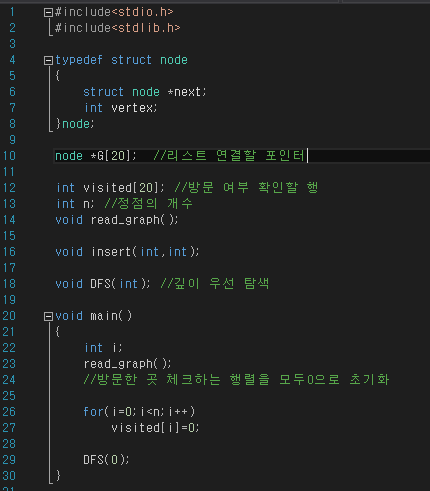
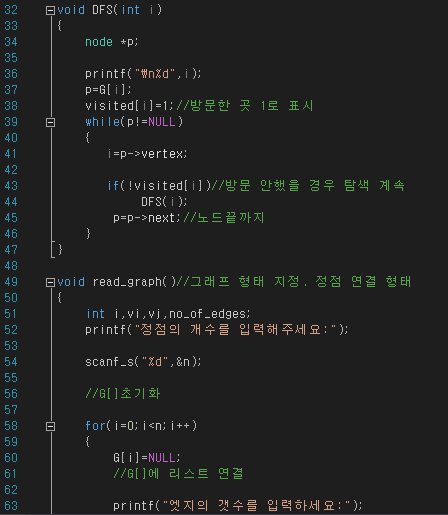
[자료구조 과제 #8]

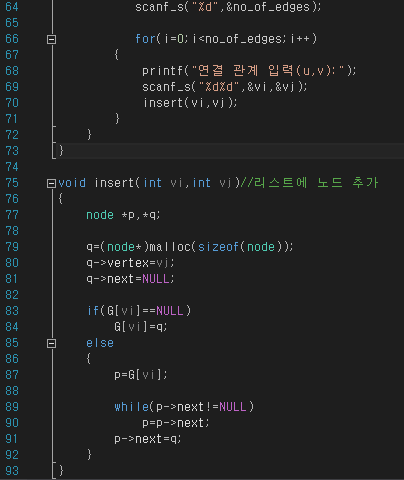
11조 한석호, 전민규, 류한길, 이예은

1. DFS

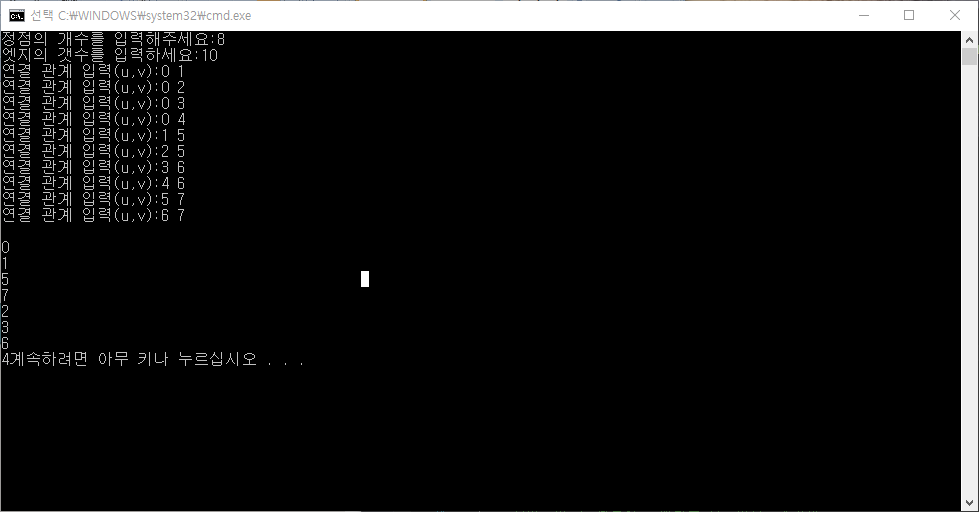
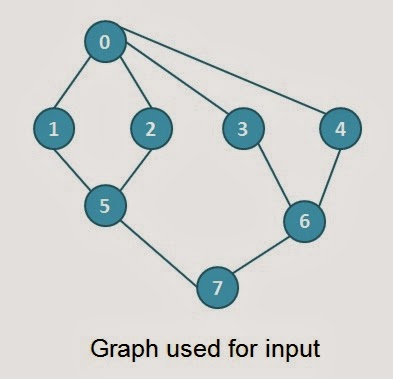
* Code







* Result

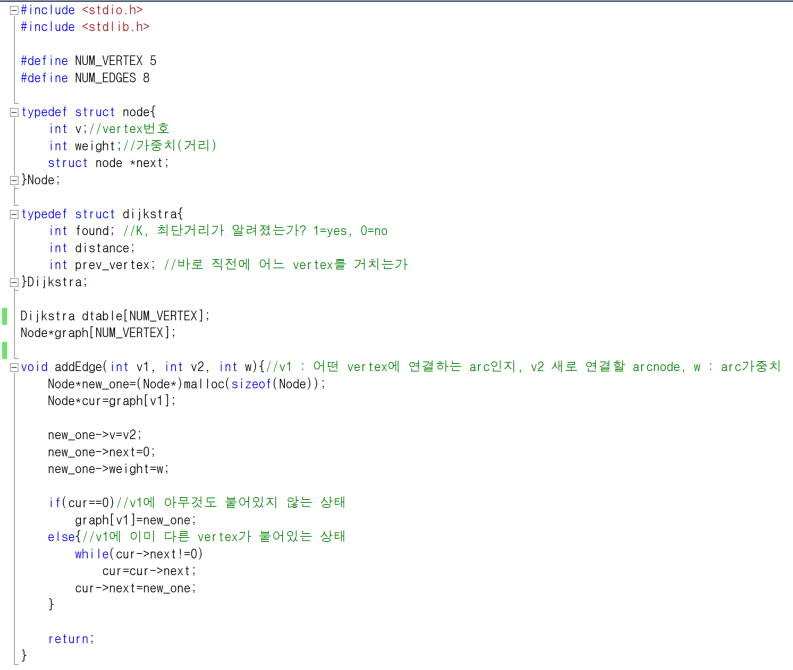


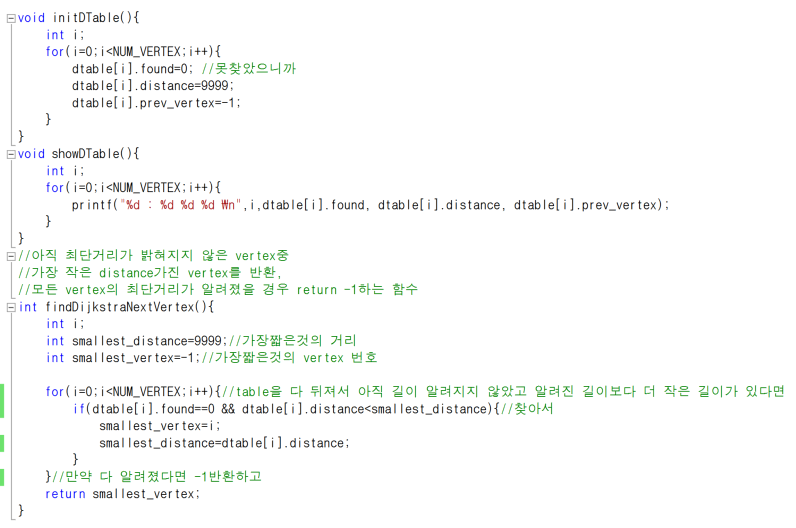
Sample Graph

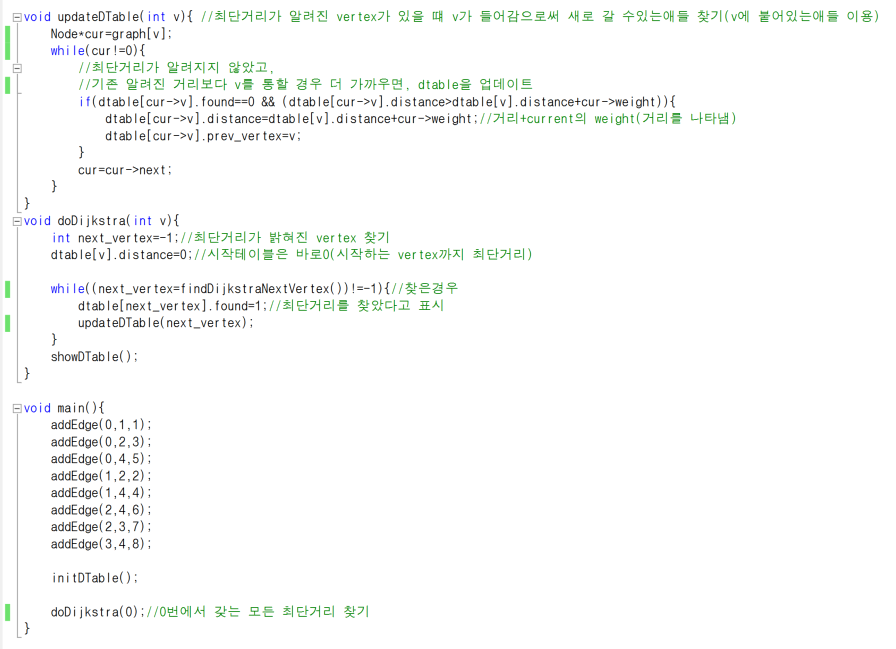
1. BFS
2. Dijkstra

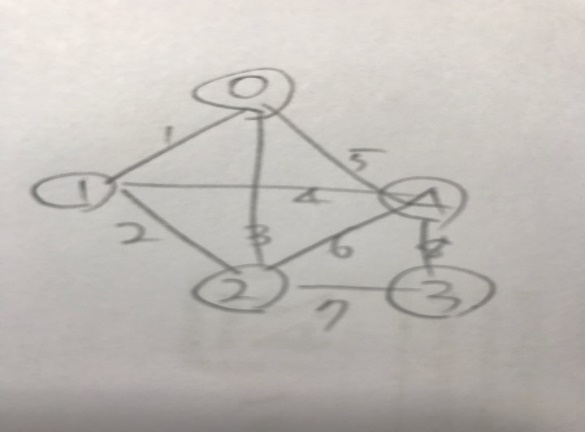
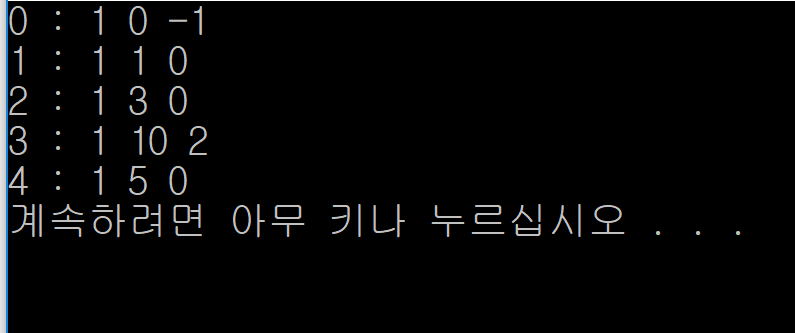
실제 팀프로젝트인 미로찾기에서 다익스트라 알고리즘을 구현하기에 앞서 구현한 다익스트라 코드입니다. 그래프는 각 vertex와 그에 연결된 arc들을 표현하기위해 포인터 배열로 전역변수로 선언하였고 다익스트라 구조체를 테이블을 표현하도록 선언하였습니다.

구현시 코드에 주석을 달아놓았습니다.





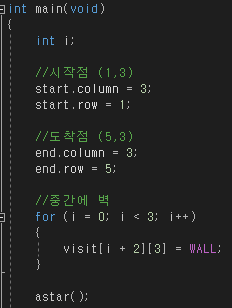




이는 다익스트라 함수 실행결과로서 ‘0’번 vertex에서 시작하여 각 vertex로의 최단거리를 나타낸 결과입니다.

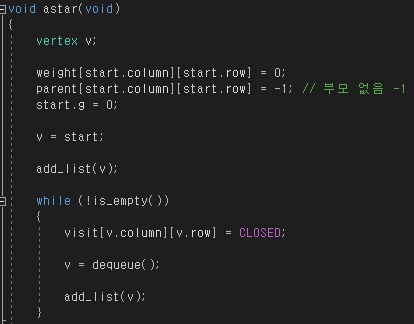
1. A\*

* Main



시작점과 도착점 그리고 벽을 설정해주고 astar() 함수를 호출합니다.

* Astar()



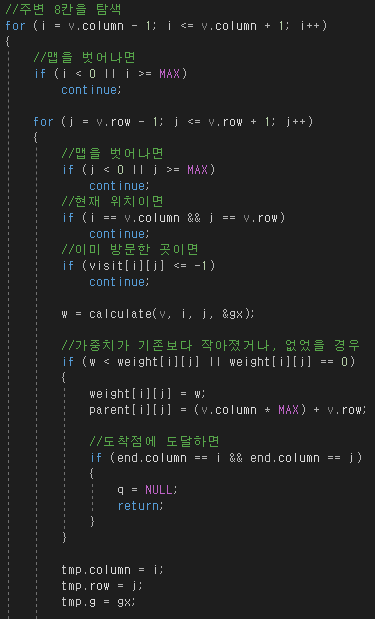
Astar() 함수에서는 시작지점을 초기화하고

Add\_list() 함수를 호출하여 시작 지점의 주변 공간을 큐에 넣습니다.

큐가 빌 때까지 주변 공간을 하나씩 불러와 큐에 넣는 방식으로 진행합니다.

(참고한 사이트에서 데이터들을 전역변수로 선언하여서 그렇게 구현했습니다만, 굳이 전역변수여야 할 필요는 없을 것 같습니다.)

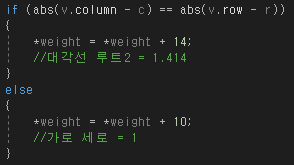
* Add\_list()



주변 8칸을 탐색하고 calculate() 함수로 가중치를 계산한 후 업데이트합니다.

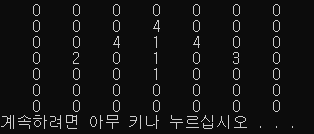
도착점이 리스트에 들어가면 큐를 비워 astar() 에서의 반복문이 끝나도록 합니다.

* Calculate()



대각선이면 루트2를 기준으로 14, 가로와 세로는 10으로 가중치를 두었습니다.

* 실행 화면



2에서 3으로 가는 길을 4로 표현하고 있습니다.

중간에 1로 표시된 벽 부분은 돌아서 진행하게 됩니다.