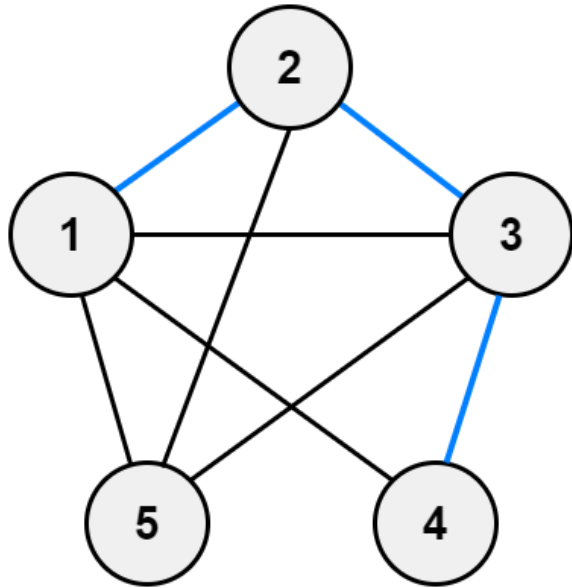


ILE3-021 그래프

정점(Node)과 간선(Edge)으로 이루어진 자료구조

간선의 방향 유무에 따라서 단방향 그래프와 무방향(양방향) 그래프로 나뉨

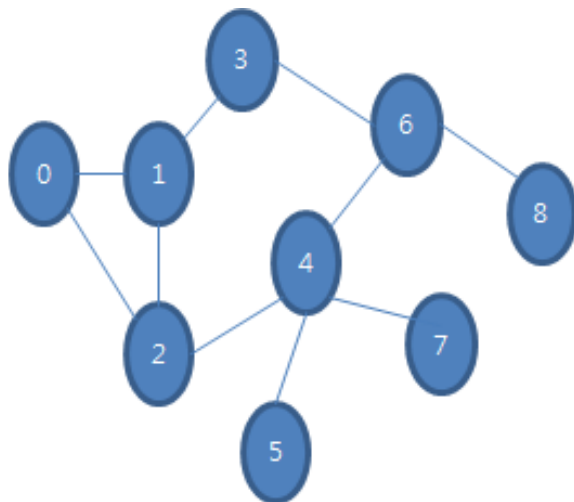


그래프의 표현 방법 ↻

인접 행렬 ↻

각 노드 간의 연결 정보를 2차원 배열을 이용하여 표현

두 노드 간의 연결 정보를 확인 시 배열을 이용하여 바로 확인 가능 $O(1)$



방향성 없는 그래프

	0	1	2	3	4	5	6	7	8
0	F	T	T	F	F	F	F	F	F
1	T	F	T	T	F	F	F	F	F
2	T	T	F	F	T	F	F	F	F
3	F	T	F	F	F	F	T	F	F
4	F	F	T	F	F	T	T	T	F
5	F	F	F	F	T	F	F	F	F
6	F	F	F	T	T	F	F	F	T
7	F	F	F	F	T	F	F	F	F
8	F	F	F	F	F	F	T	F	F

인접 행렬로 표현

인접 리스트

리스트를 이용하여 연결된 요소들에 대해서 연결 정보를 저장
트리와 유사한 방식으로 저장을 수행하며, 루프를 허용

두 노드 간의 연결 정보를 확인 시 노드의 순회가 필요 $O(n)$

그래프의 노드 순회 방법

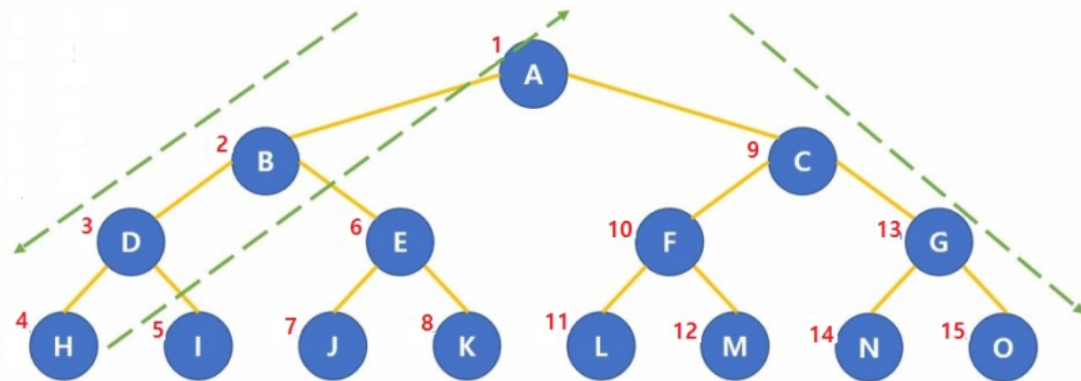
그래프의 각 노드를 읽어오는 방법

그래프를 활용한 알고리즘에서 문제 해결 과정에서 사용

깊이 우선 탐색(Depth First Search)

개별 경로를 끝까지 탐색하는 방법

- 하나의 노드를 선택한다
 - 다른 노드의 연결이 있는 경우 다른 노드로 이동
 - 노드가 추가적인 연결이 없거나 Loop가 발생하는 경우, 이전 노드로 이동
- 위의 과정을 이동할 노드가 없을 때까지 반복



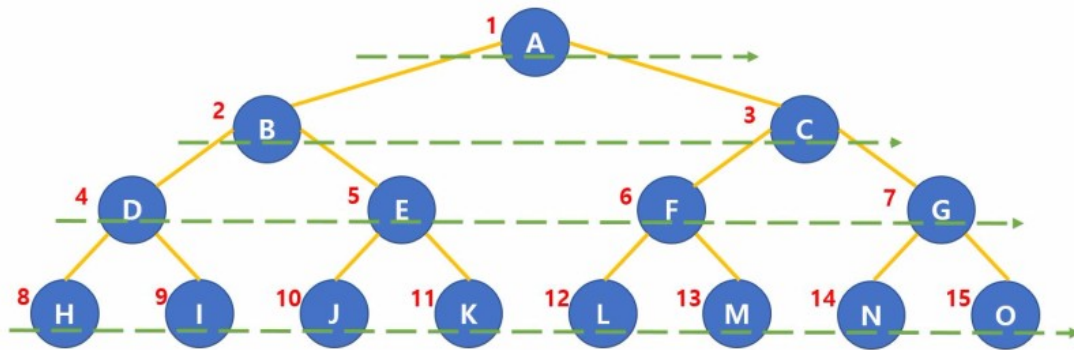
DFS 탐색 순서

<http://blog.naver.com/occidere>

너비 우선 탐색 (Breadth First Search)

특정 노드와 연결된 모든 노드를 확인하고 다음 단계로 넘어간다

- 하나의 노드를 선택한다
 - 노드와 연결된 노드가 있는 경우 방문 후 다시 이전 노드로 복귀
 - 위 과정을 방문하지 않은 노드가 있을 때 까지 반복
- 다음 노드로 이동한다
 - 위 과정을 반복하고, 다음 노드가 없는 경우 이전 노드로 복귀
- 더이상 진행 할 노드가 없을 때까지 반복



BFS 탐색 순서

<http://blog.naver.com/occidere>

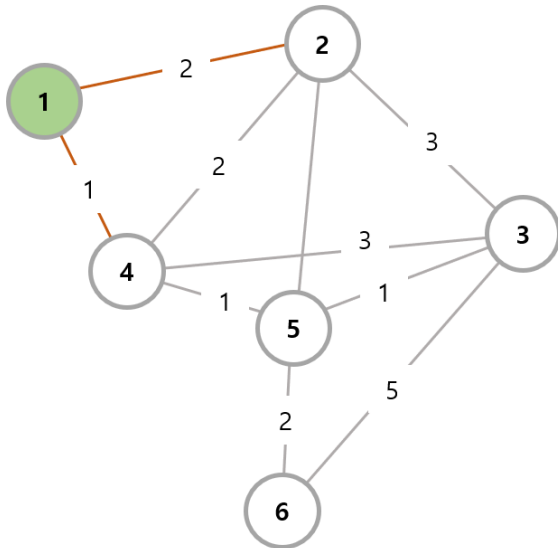
그래프의 활용

최단 거리 탐색

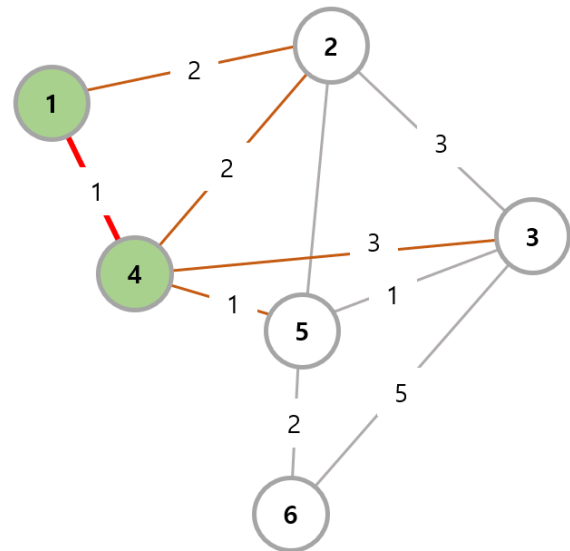
다익스트라(Dijkstra) 알고리즘을 이용하여 특정 노드에서 이동 시의 최단 비용을 가지는 그래프 생성

후보 노드를 선택하고, 해당 엣지에 대한 가중치가 가장 작은 값을 선택 (BFS 응용)

1. 시작 노드 선택 후 후보 엣지 선택

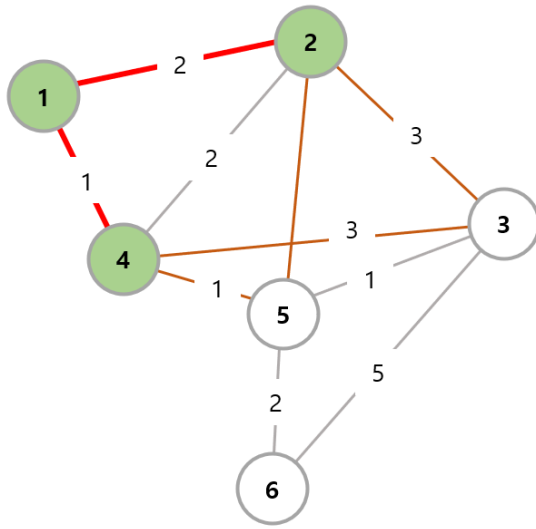


2. 후보 엣지 중 가장 낮은 가중치를 선택하여 노드 4를 선택하고 노드 4와 연결된 엣지를 후보 엣지로 선택

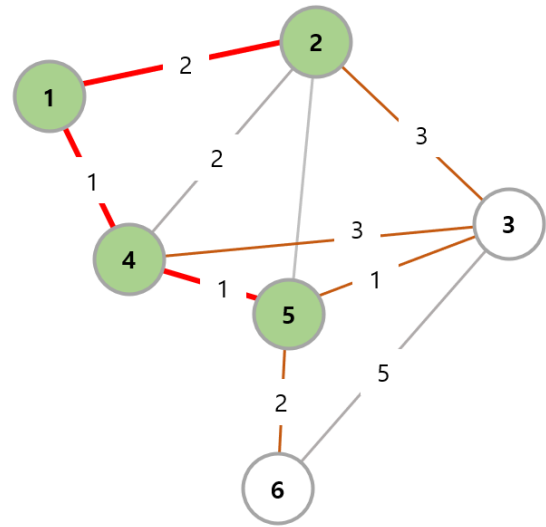


3. 후보 엣지 중 가장 낮은 가중치를 선택하여 노드 2를 선택하고 노드 2와 연결된 엣지를 후보 엣지로 선택
그리고 Loop를 발생할 수 있는 엣지는 후보 엣지에서 제외

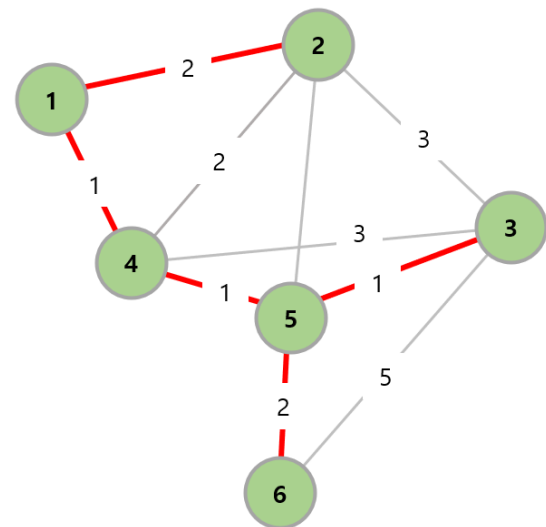
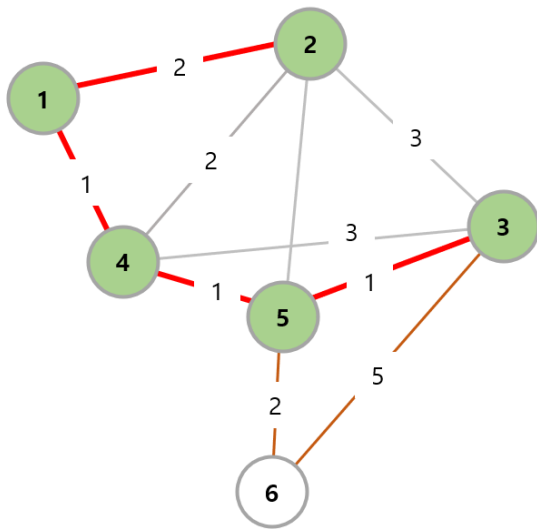
4. 후보 엣지 중 가장 낮은 가중치를 선택하여 노드 5를 선택하고 노드 5와 연결된 엣지를 후보 엣지로 선택
그리고 Loop를 발생할 수 있는 엣지는 후보 엣지에서 제외



5. 후보 엣지 중 가장 낮은 가중치를 선택하여 노드 3를 선택하고 노드 3와 연결된 엣지를 후보 엣지로 선택
그리고 Loop를 발생할 수 있는 엣지는 후보 엣지에서 제외



5. 후보 엣지 중 가장 낮은 가중치를 선택하여 노드 3를 선택
더이상 다음 노드가 없으므로 종료



과제

다익스트라 알고리즘을 구현하고 위의 예제의 그래프에 대해서 적용