

씨앗 senior

Week 5

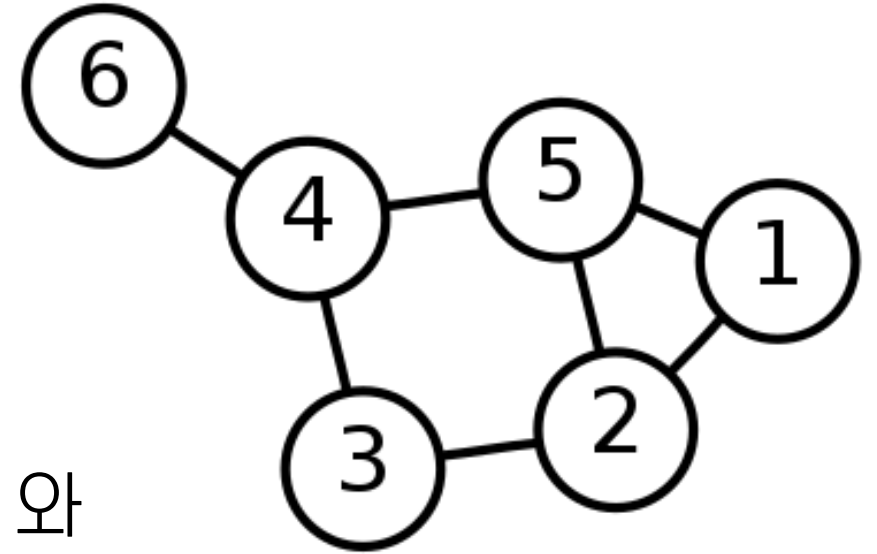
Date : 23.05.25

Made by 은재민

오늘의 진도

1. 그래프에 대한 이해
2. 트리
3. 그래프 탐색
4. BFS 와 DFS 알고리즘
5. 탐색알고리즘 활용
6. 예제

그래프의 정의



그래프 $G = (V, E)$:

어떤 자료나 개념을 표현하는 정점들의 집합 V 와
이들을 연결하는 간선들의 집합 E 으로 구성된 자료구조

정점(vertex/node) : 여러가지 특성을 가질 수 있는 개체

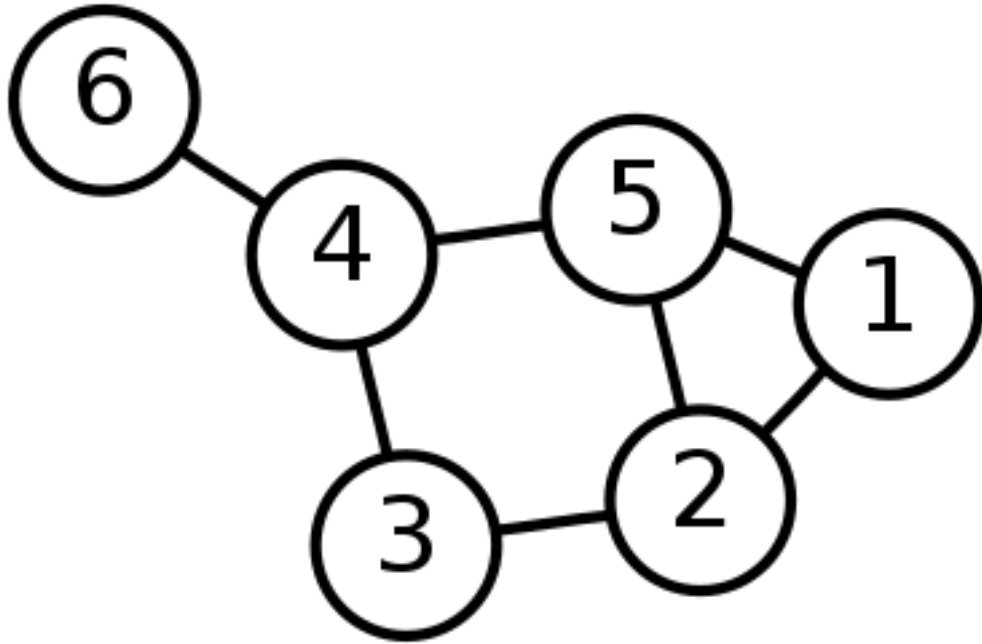
간선(edge) : 정점과 정점 사이의 관계

차수(degree) : 인접한 정점의 개수

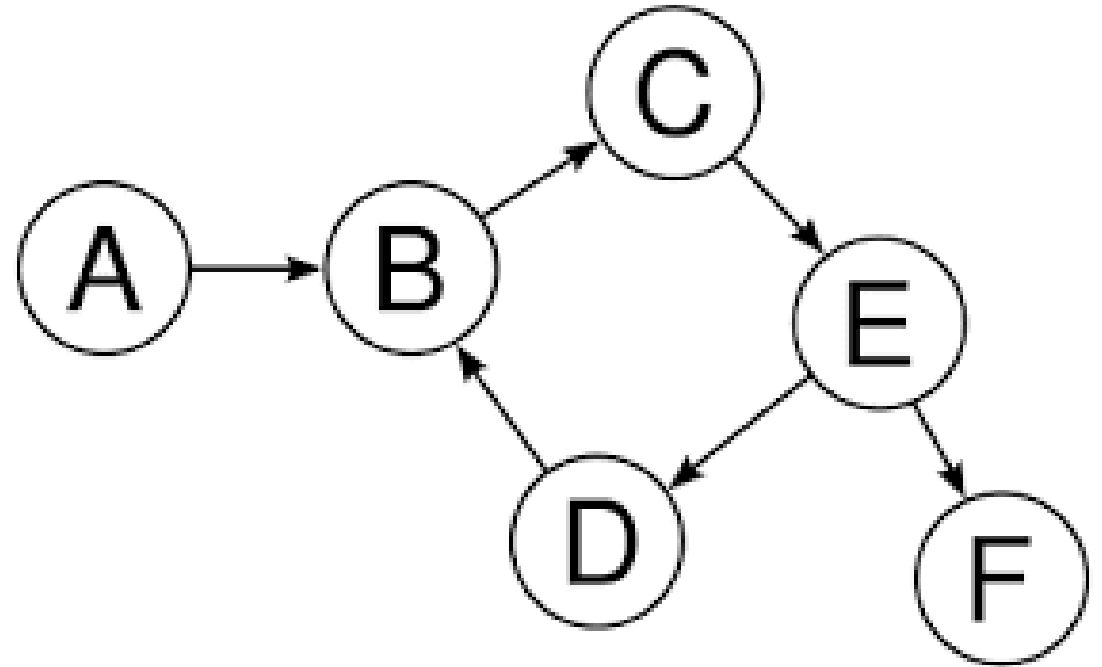
사이클(cycle) : V_1 에서 출발해서 다시 V_1 으로 돌아올 수 있는 경로

그래프의 종류

1. 방향의 유무



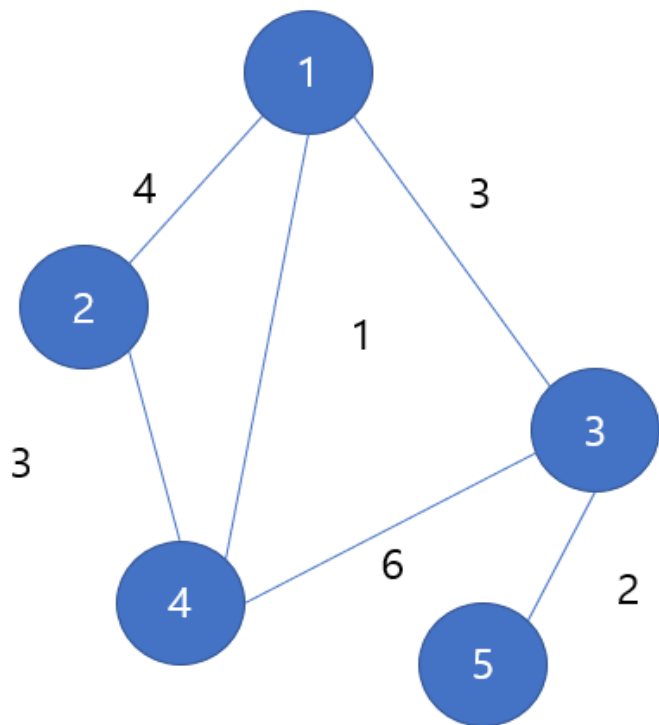
무방향 그래프



방향 그래프

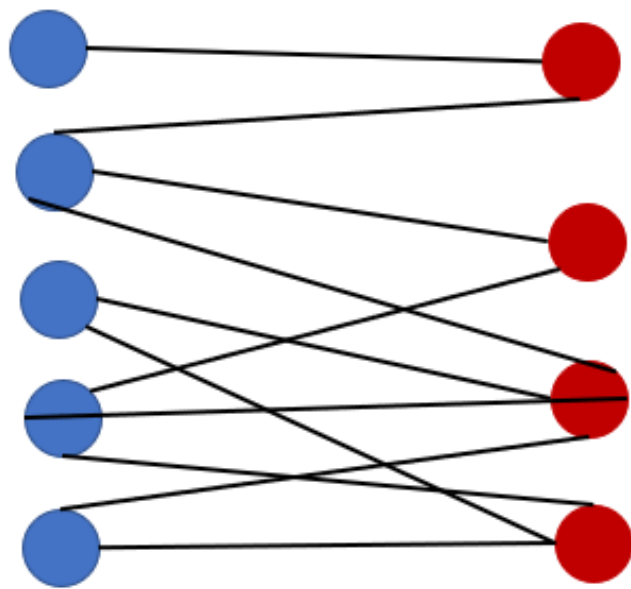
그래프의 종류

2. 가중치의 유무

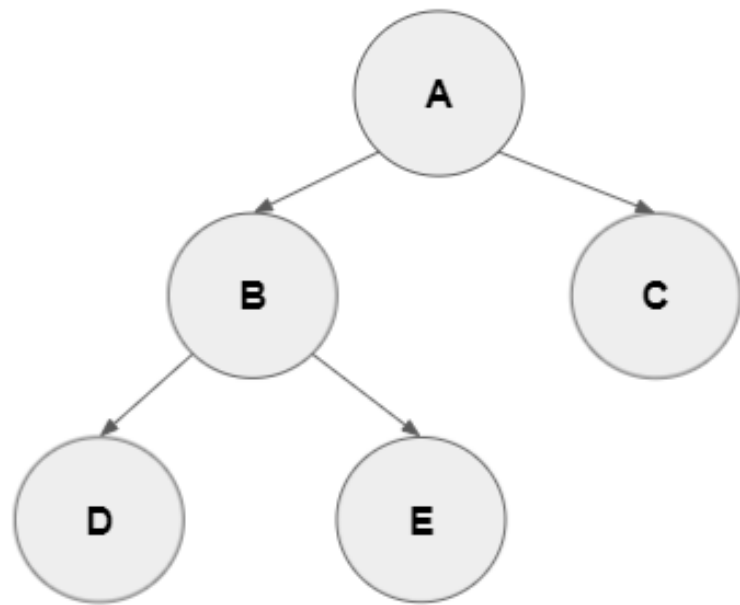


가중치 그래프

3. 특성에 따라

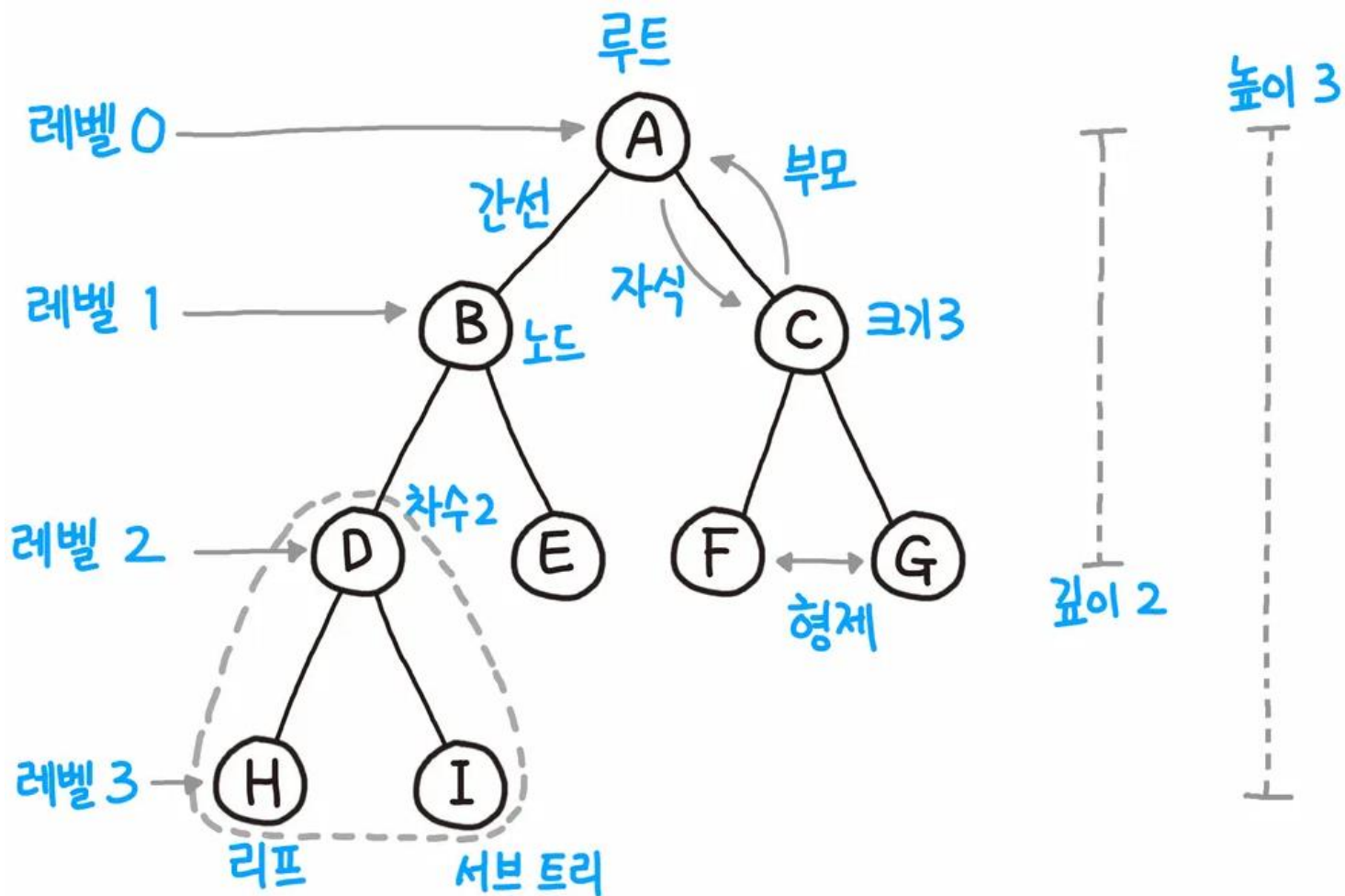


이분 그래프



트리

트리?



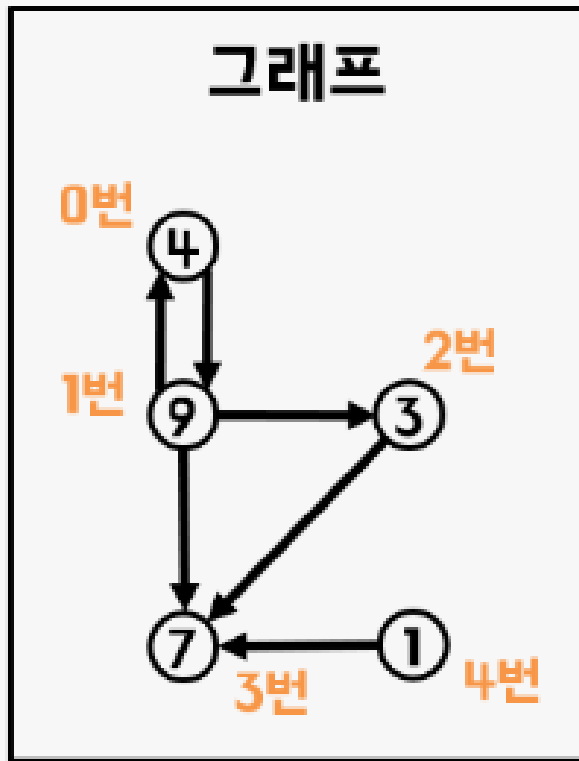
트리 : 그래프의 한 종류

트리의 특징

- 계층구조
- 1개의 최상위 노드 : 루트
- 부모/자식/형제 관계
- 사이클 X
- 자식은 1개의 부모를 가짐
- 간선의 개수 : 노드의 개수 - 1

그래프의 표현

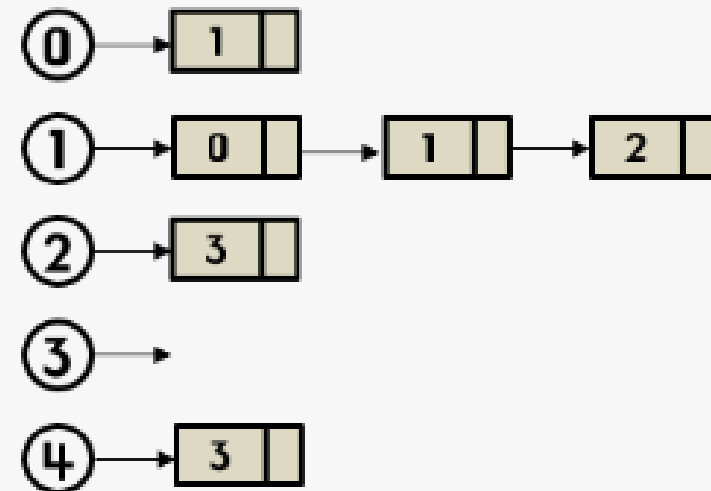
추상적인 개념인 그래프를 프로그래밍 언어로 표현하는 것.



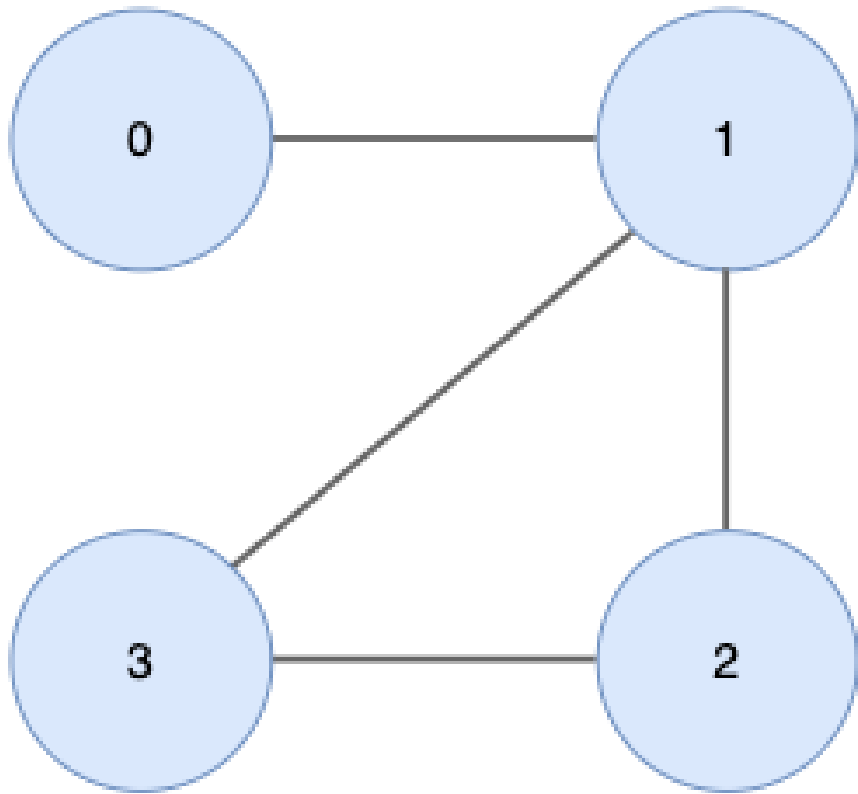
1) 인접 행렬 표현

	0	1	2	3	4
0	0	1	0	0	0
1	1	0	1	1	0
2	0	0	0	1	0
3	0	0	0	0	0
4	0	0	0	1	0

2) 인접 리스트 표현



인접 행렬로 무방향 그래프 표현

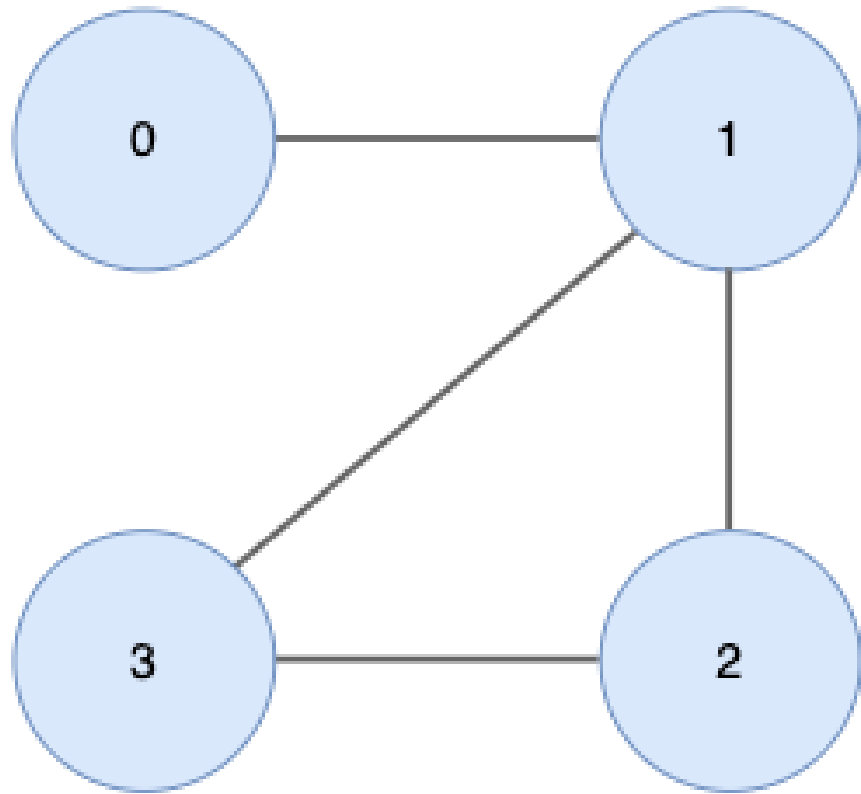


Undirected Graph

	0	1	2	3
0	0	1	0	0
1	1	0	1	1
2	0	1	0	1
3	0	1	1	0

Adjacency Matrix

인접 행렬로 무방향 그래프 표현

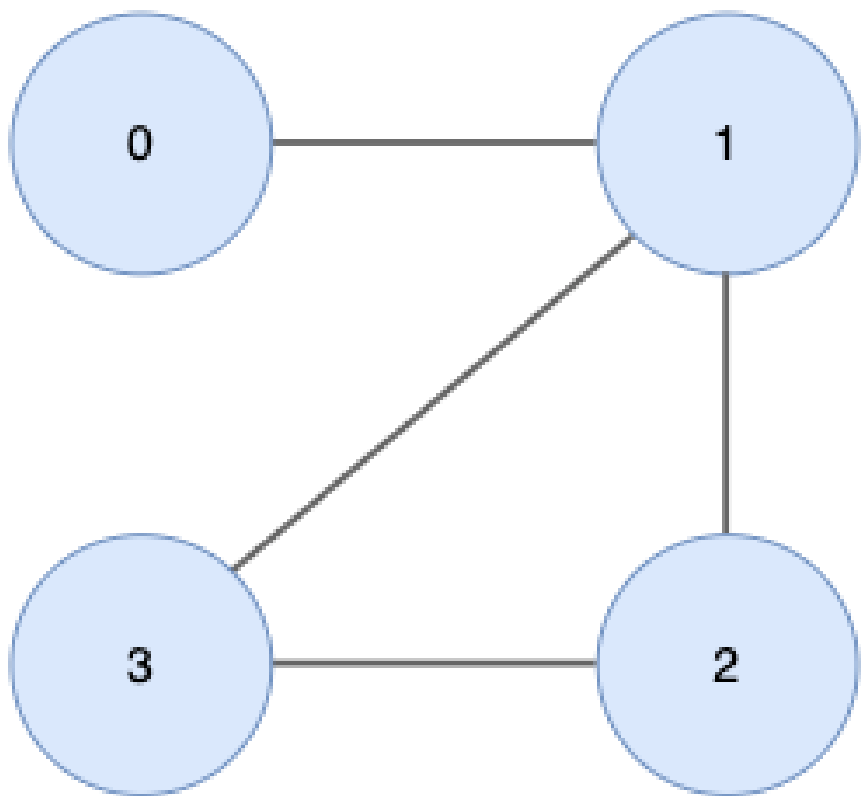


Undirected Graph

노드 A에서 노드A로
가는 간선은없으므로
(a,a)는 모두 0으로 표현

	0	1	2	3
0	0	1	0	0
1	1	0	1	1
2	0	1	0	1
3	0	1	1	0

Adjacency Matrix



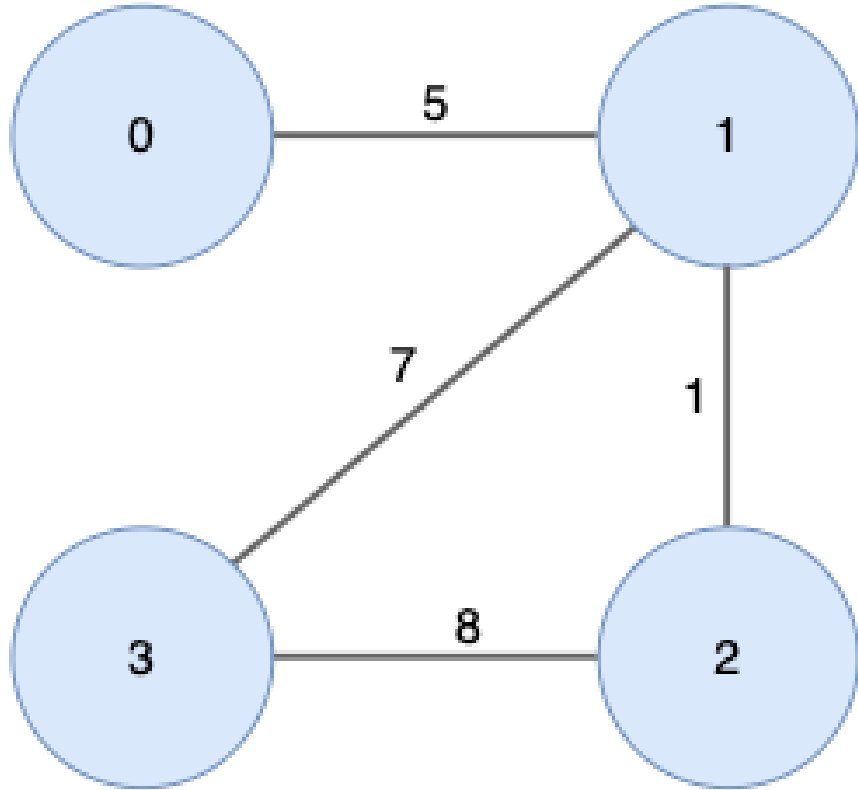
Undirected Graph

무방향 그래프의
인접행렬은 대칭구조

	0	1	2	3
0	0	1	0	0
1	1	0	1	1
2	0	1	0	1
3	0	1	1	0

Adjacency Matrix

인접 행렬로 가중치 그래프 표현

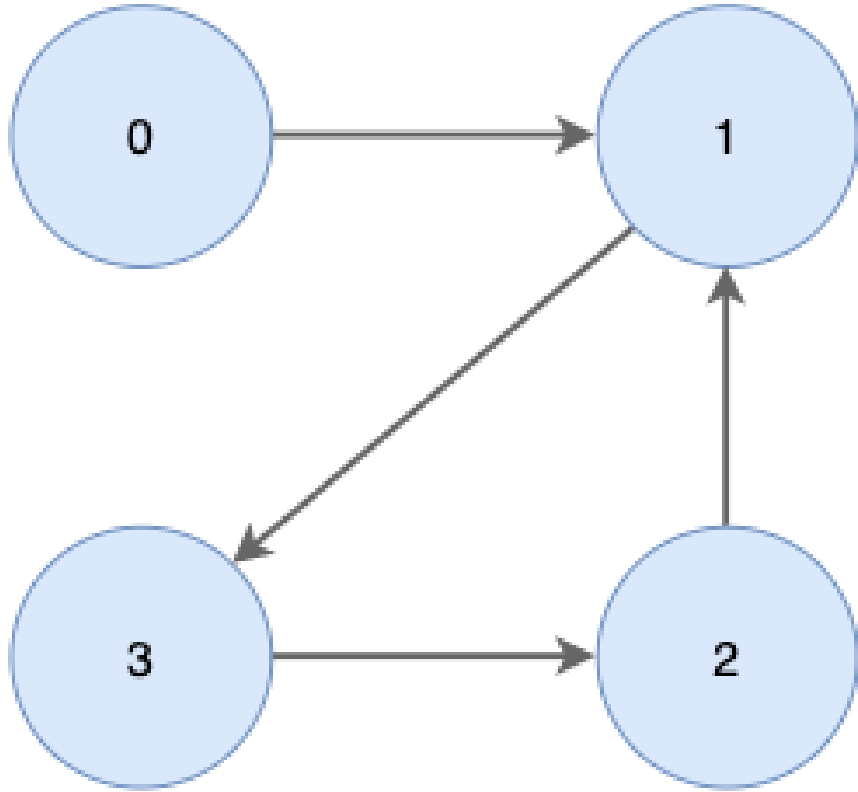


Weighted Graph

	0	1	2	3
0	0	5	0	0
1	5	0	1	7
2	0	1	0	8
3	0	7	8	0

Adjacency Matrix

인접 행렬로 방향 그래프 표현



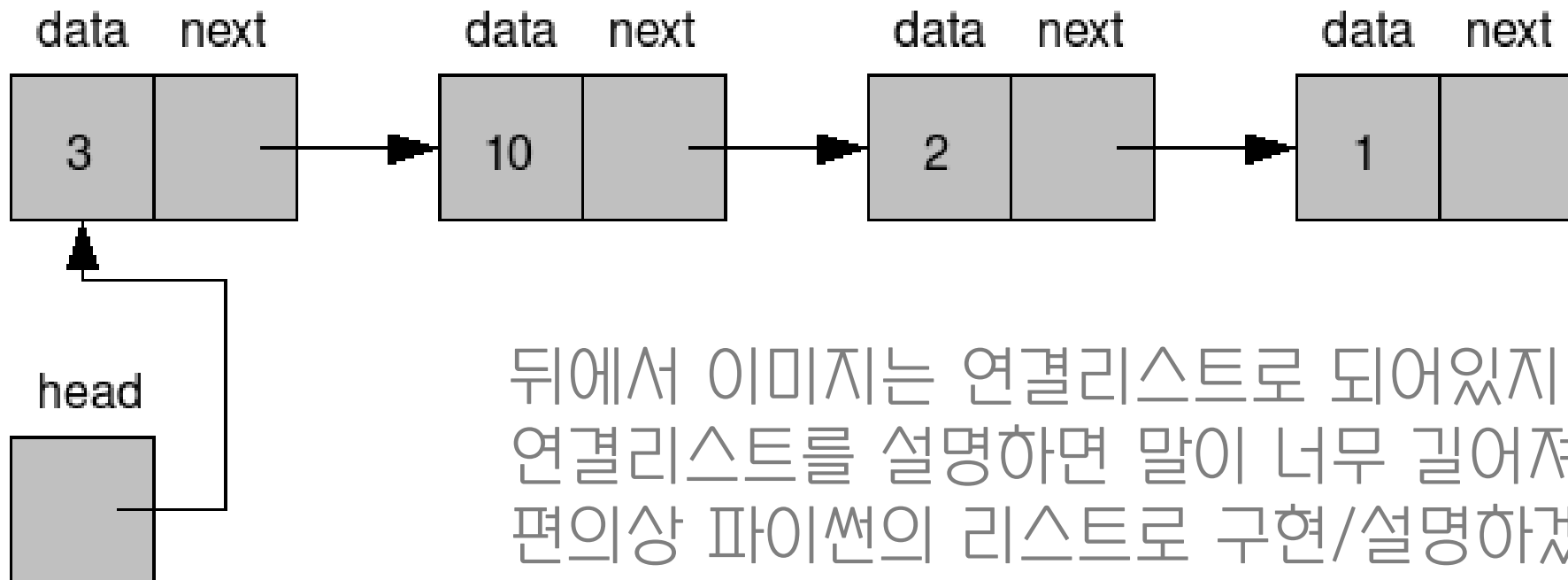
Directed Graph

	0	1	2	3
0	0	1	0	0
1	0	0	0	1
2	0	1	0	0
3	0	0	1	0

Adjacency Matrix

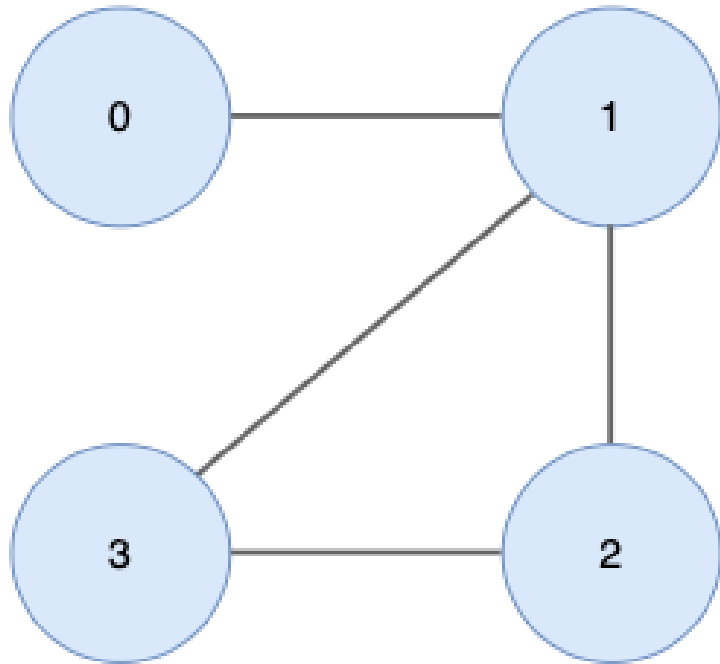
인접 리스트로 표현

연결 리스트(linked list)?

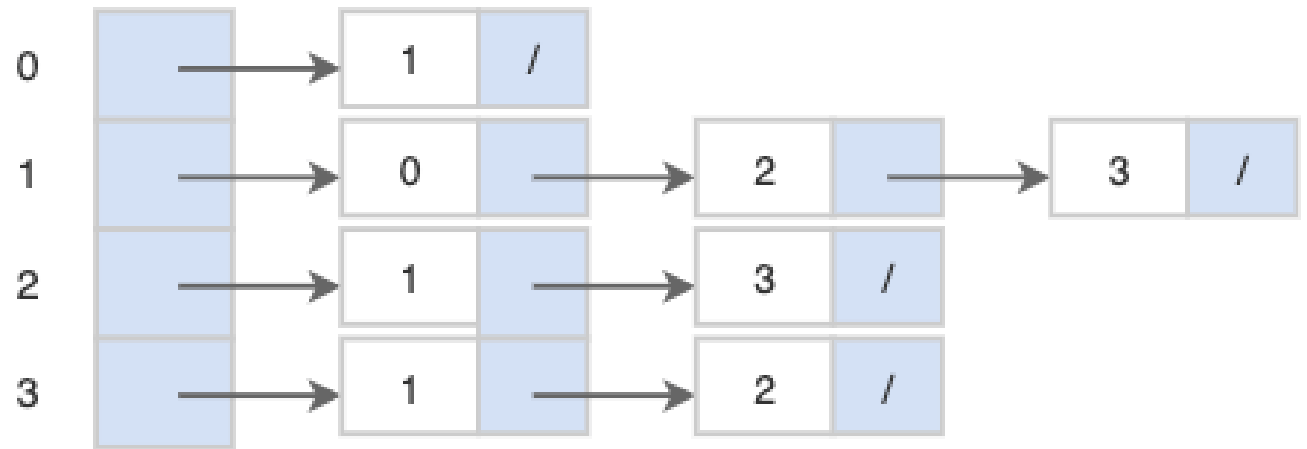


뒤에서 이미지는 연결리스트로 되어있지만
연결리스트를 설명하면 말이 너무 길어져서
편의상 파이썬의 리스트로 구현/설명하겠습니다.

인접 리스트로 무방향 그래프 표현

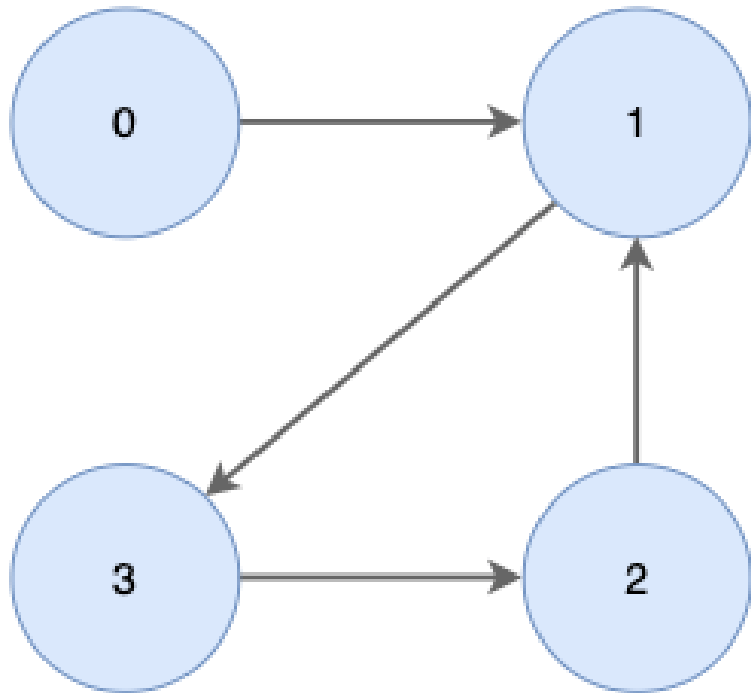


Undirected Graph

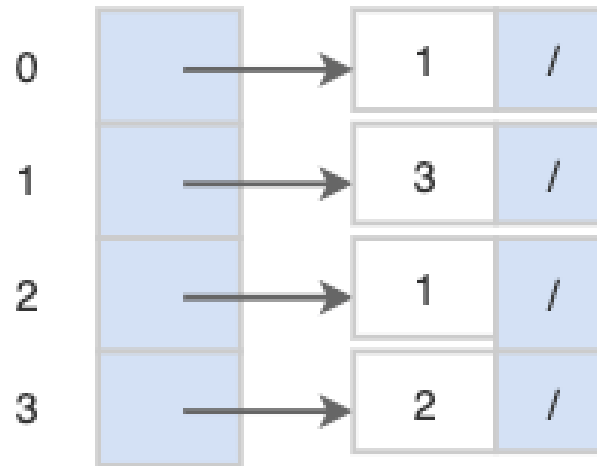


Adjacency List

인접 리스트로 방향 그래프 표현

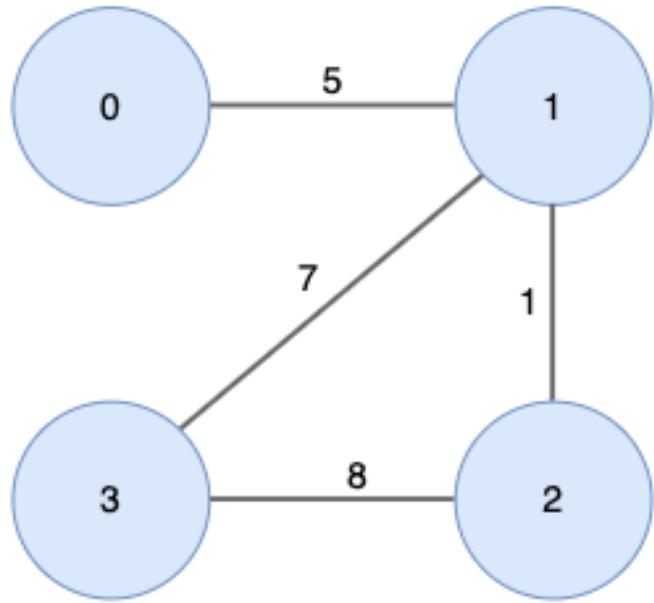


Directed Graph

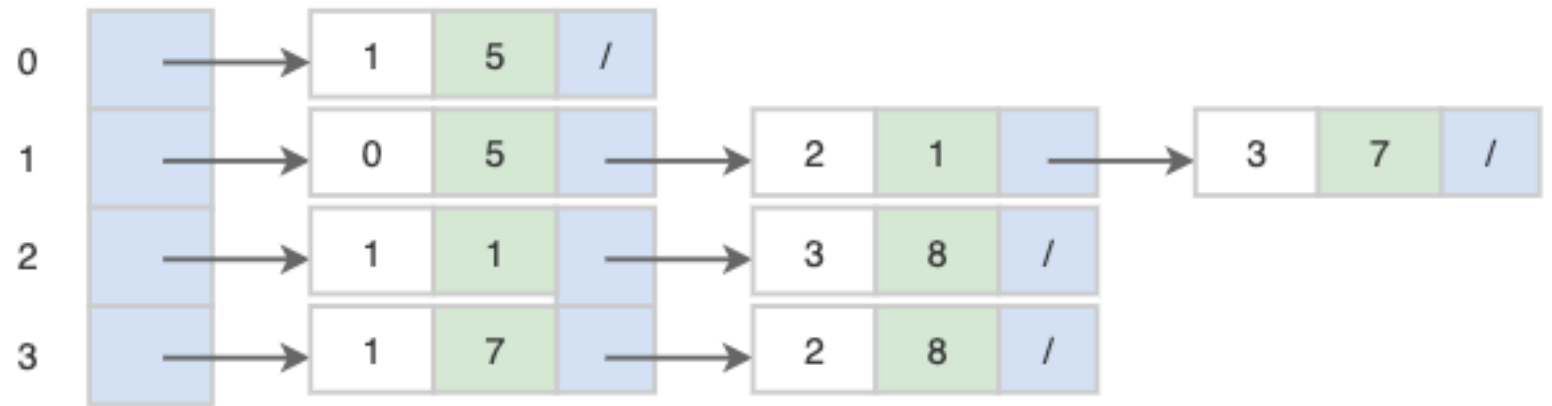


Adjacency List

인접 리스트로 가중치 그래프 표현



Weighted Graph



Adjacency List

가중치를 나타내기 위해
데이터 부분을 두개로 나눔
데이터 : 노드번호 / 가중치

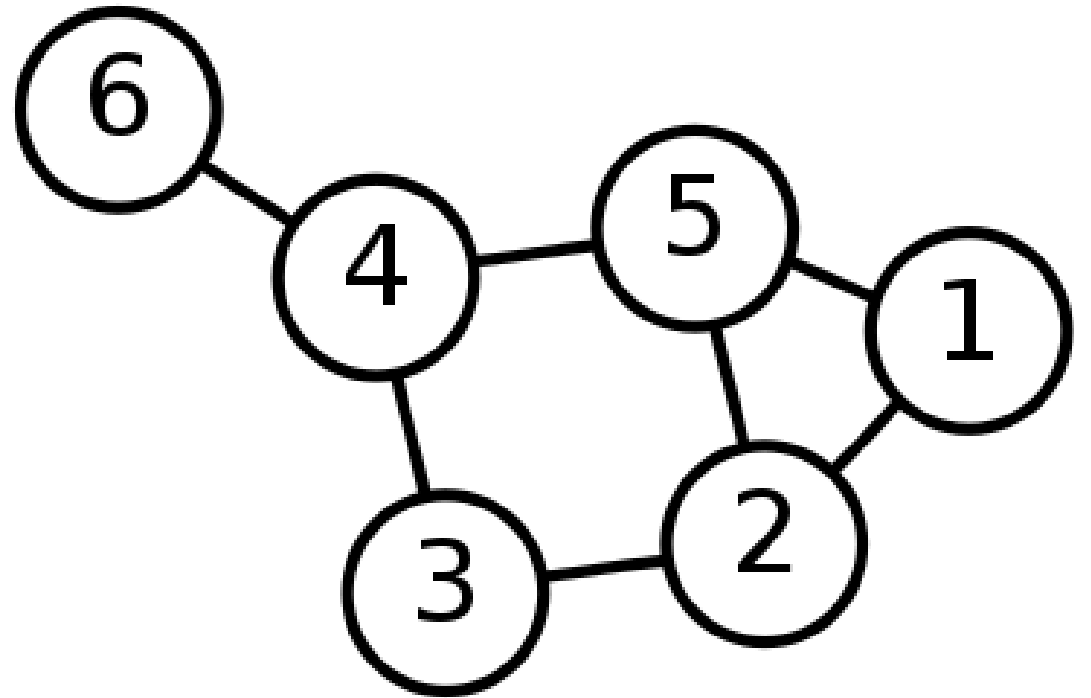
코드 작성 예시

- 인접 행렬 표현법

[Senior/week5/Adj_Matrix.py](#)

- 인접 리스트 표현법

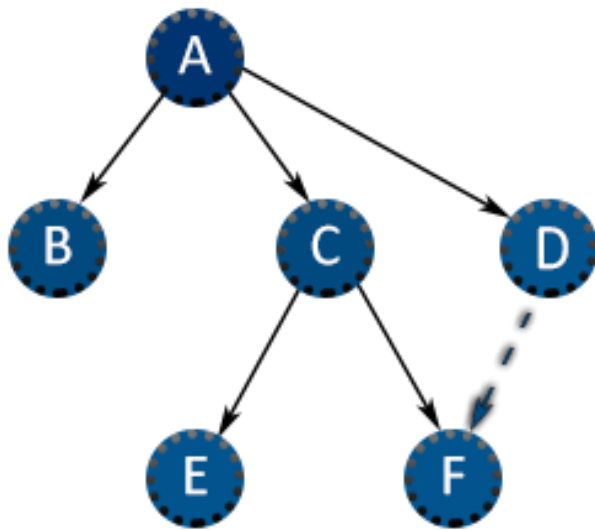
[Senior/week5/Adj_List.py](#)



그래프의 탐색

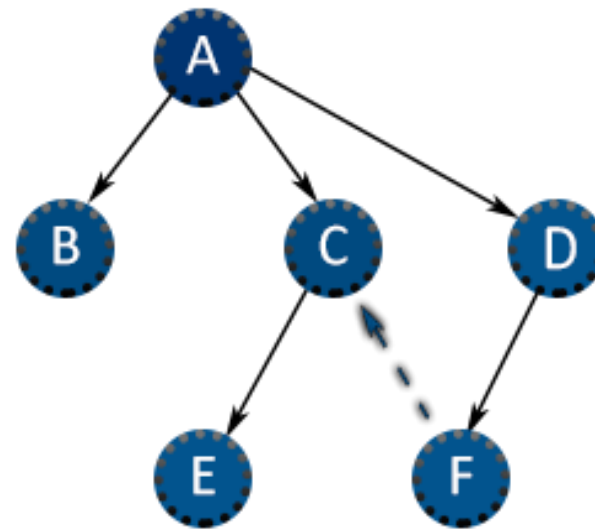
그래프의 정점들을 간선을 통해 탐색하는 것 ex) BFS, DFS

BFS



A B C D E F

DFS



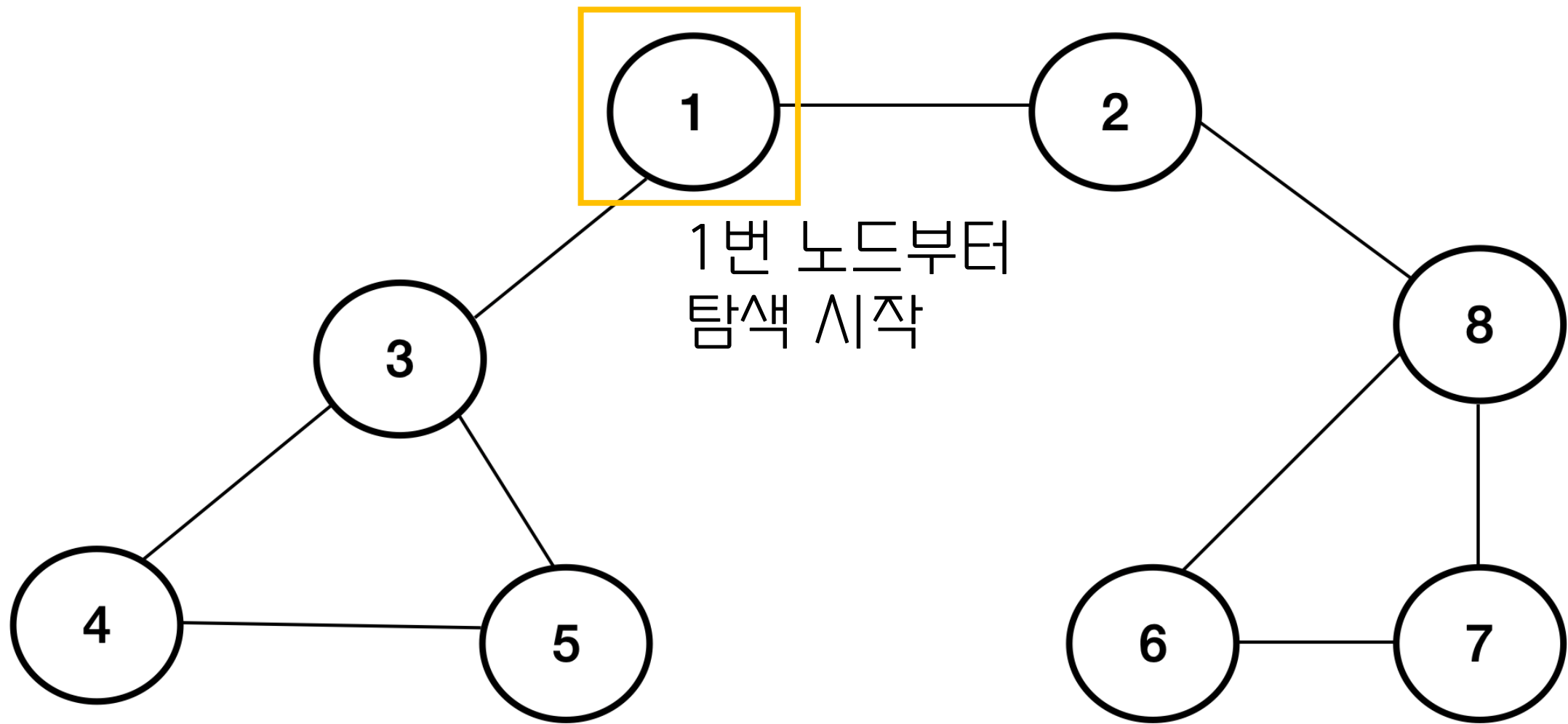
A D F C E B

너비우선탐색 Breadth First Search

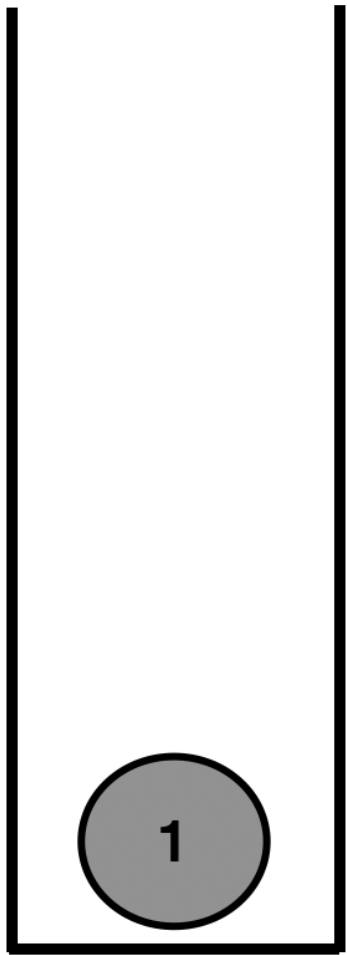
하나의 노드를 루트로 정하고 가까운 정점부터 탐색하는 방법
= 넓게 탐색하는 방법

선입선출 자료구조, Queue를 이용하여 구현한다.

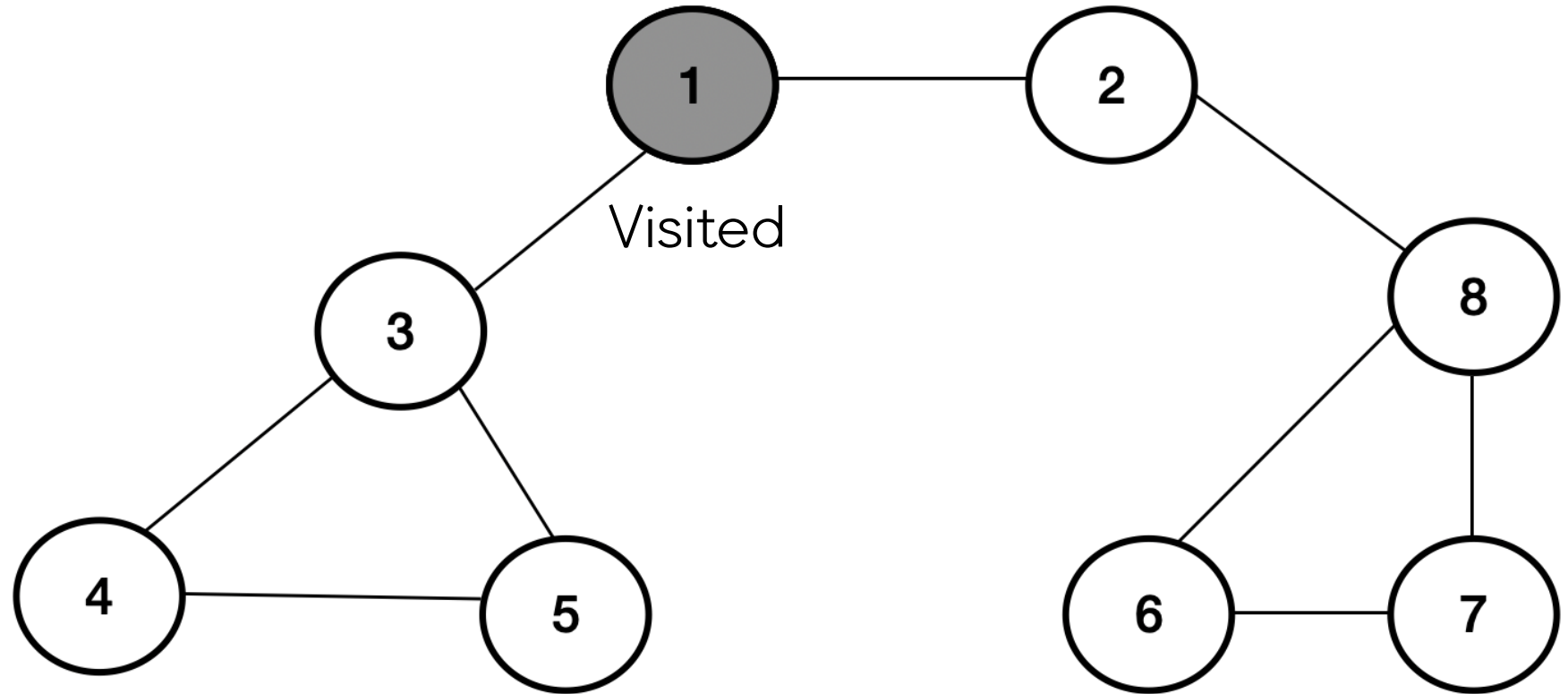
1. 큐에서 노드 A를 꺼냄
2. 노드A 에 인접한 노드 B,C,... 를 큐에 삽입
3. 1~2 반복.



루트노드를 큐에 삽입, 방문처리



큐

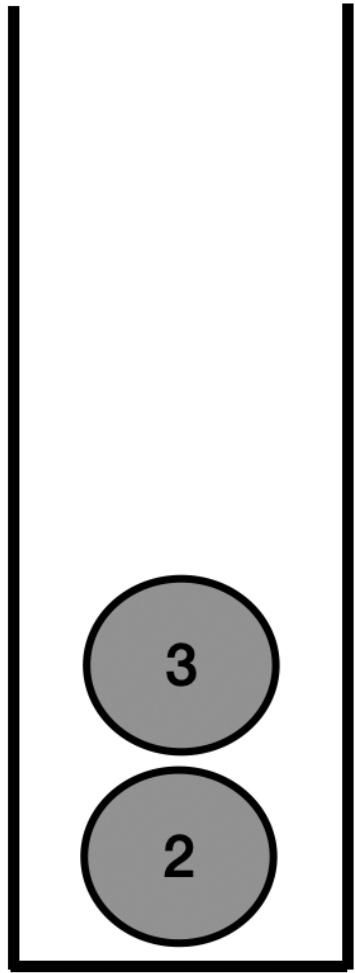


<https://heytech.tistory.com/>

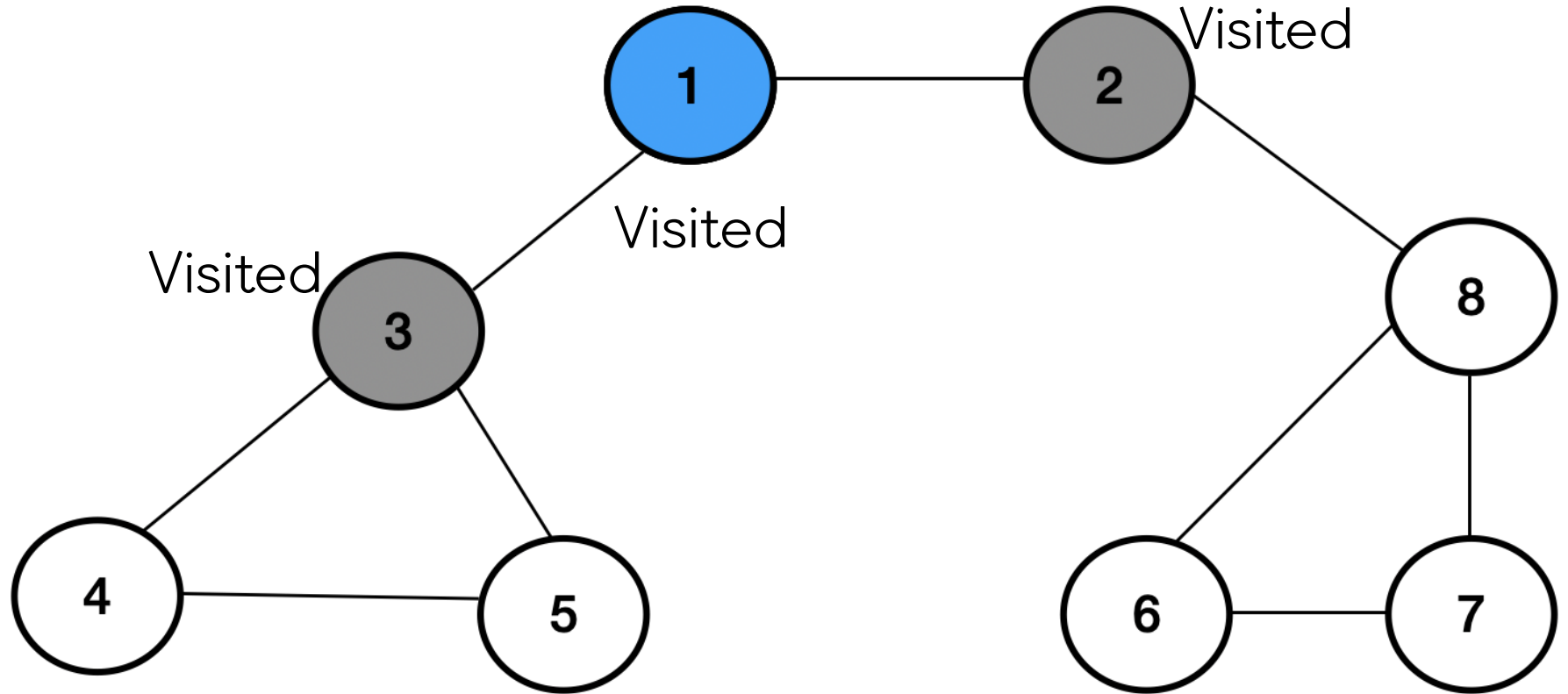
현재 방문 노드 : 1

큐에서 노드 하나를 꺼내서 인접 노드 확인 : 2,3번

인접한 노드를 큐에 삽입, 방문처리



큐

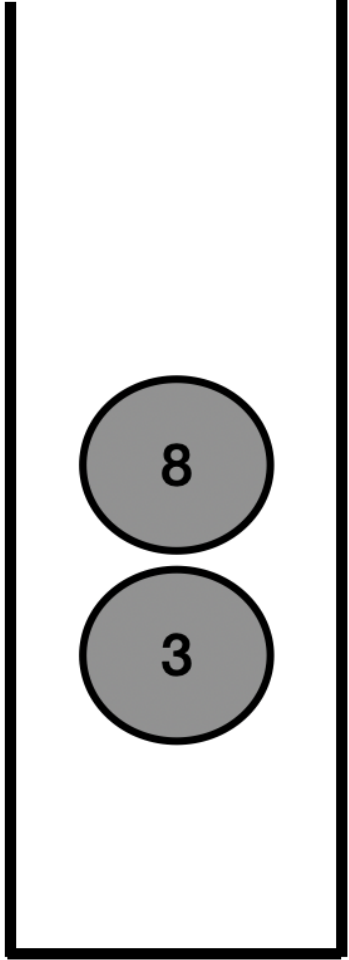


<https://heytech.tistory.com/>

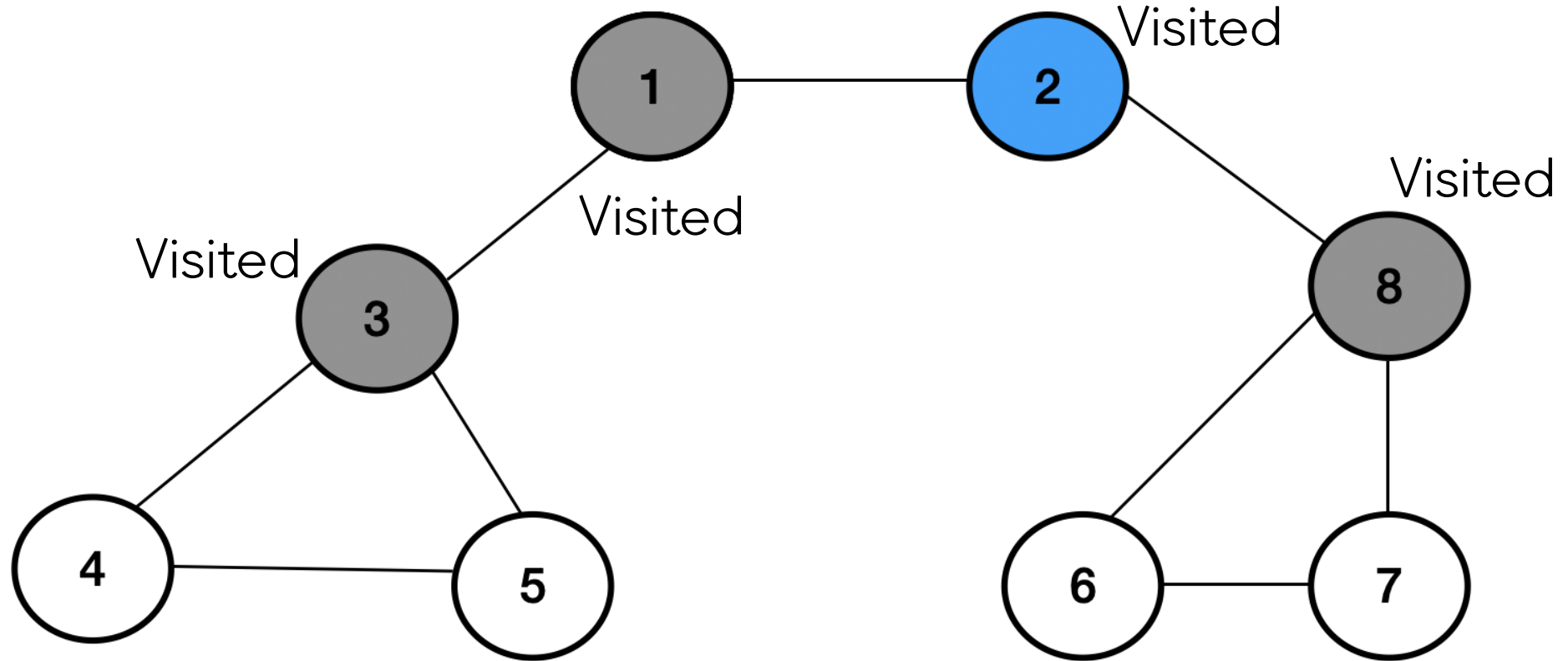
현재 방문 노드 : 1-2-3

큐에서 노드 하나를 꺼내서 인접 노드 확인 : 1,8번

인접한 노드 중 방문하지 않은 노드를 큐에 삽입, 방문처리



큐

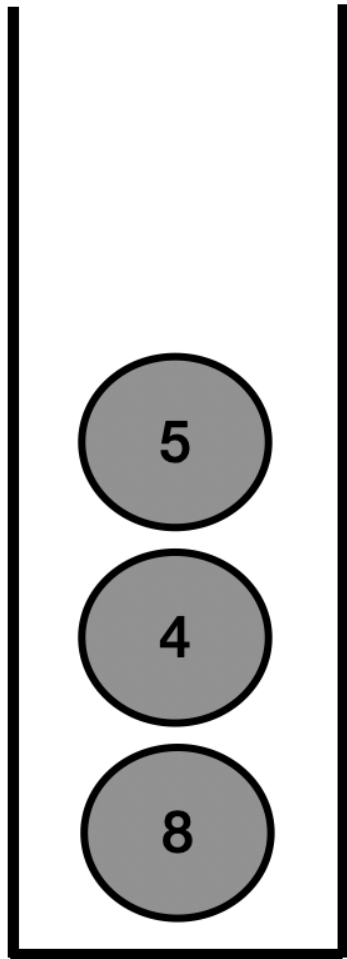


<https://heytech.tistory.com/>

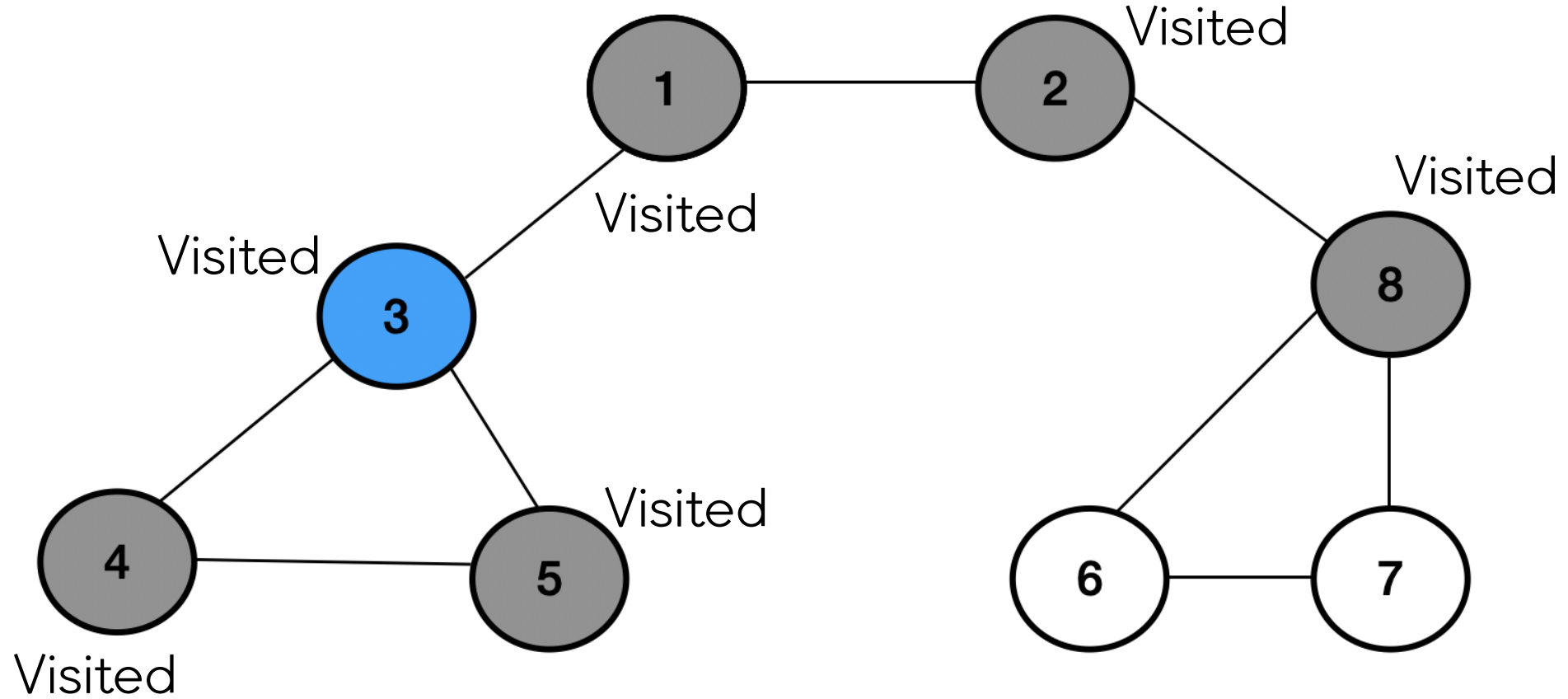
현재 방문 노드 : 1-2-3-8

큐에서 노드 하나를 꺼내서 인접 노드 확인 : 1,4,5번

인접한 노드 중 방문하지 않은 노드를 큐에 삽입, 방문처리



큐

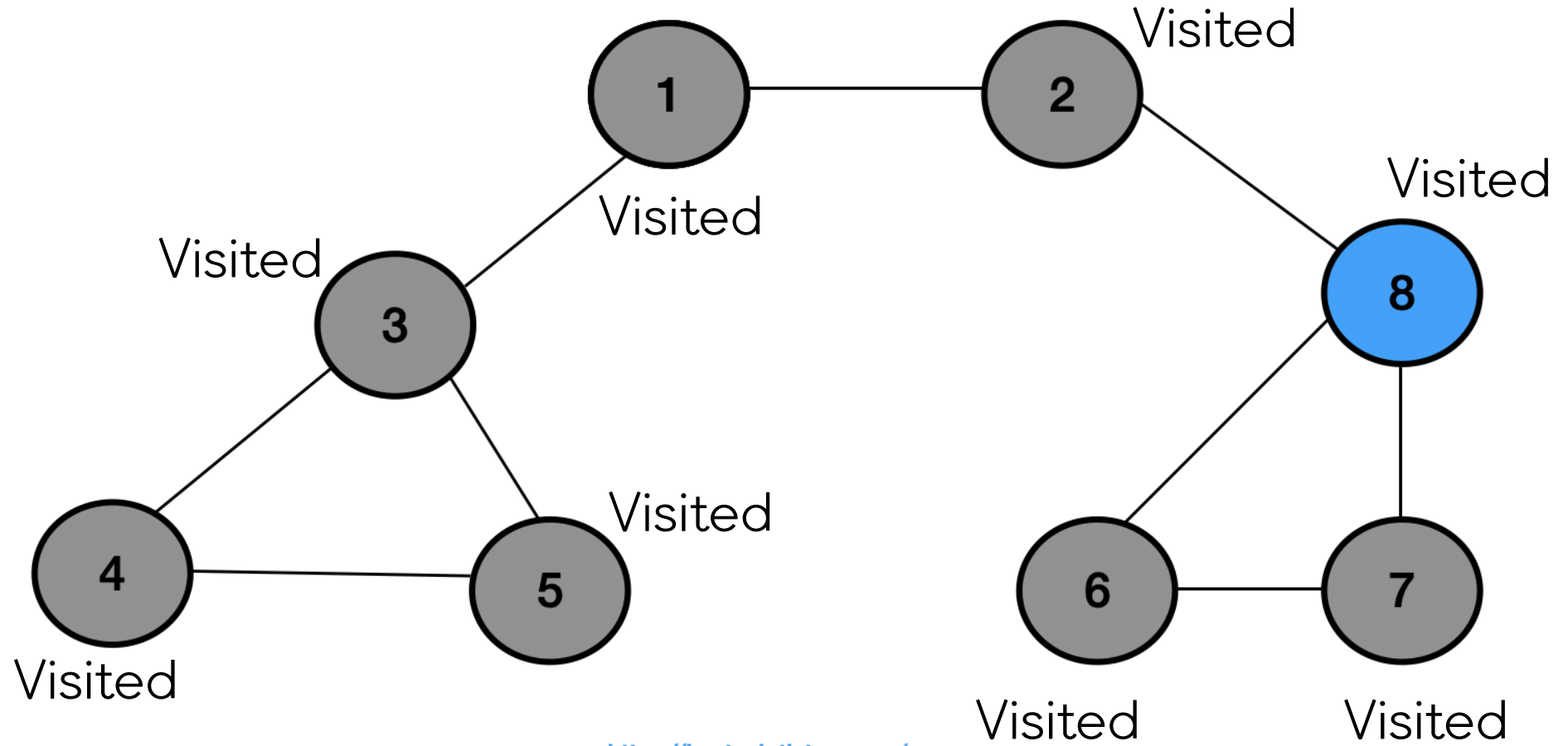
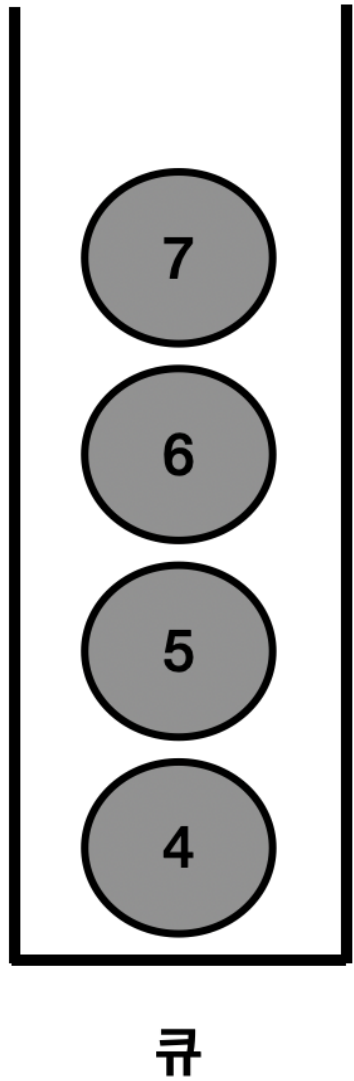


<https://heytech.tistory.com/>

현재 방문 노드 : 1-2-3-8-4-5

큐에서 노드 하나를 꺼내서 인접 노드 확인 : 2,6,7번

인접한 노드 중 방문하지 않은 노드를 큐에 삽입, 방문처리

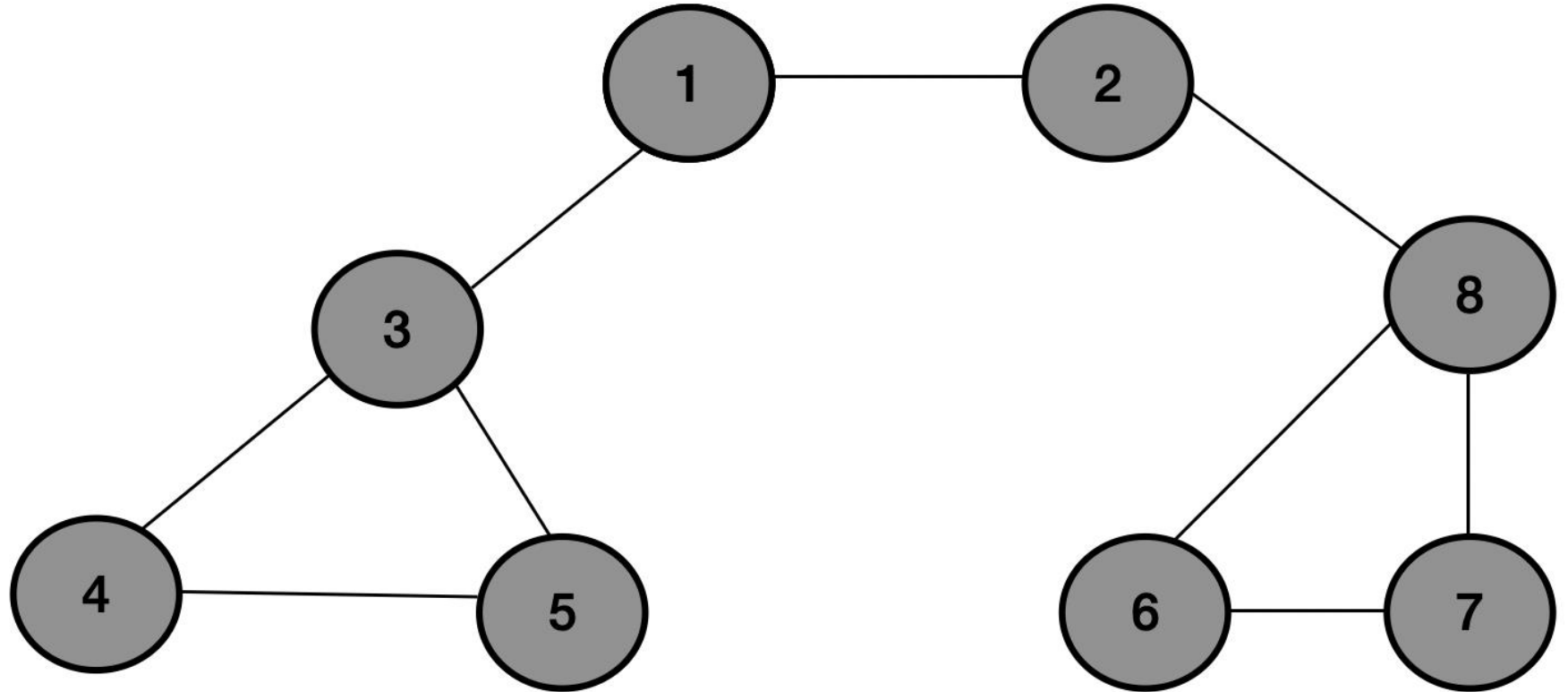


<https://heytech.tistory.com/>

현재 방문 노드 : 1-2-3-8-4-5-6-7

4, 5, 6, 7을 차례로 꺼내는 동안 새로운 삽입 없음
(인접하면서 방문하지 않은 노드X)

큐가 비면 탐색 종료



큐

<https://heytech.tistory.com/>

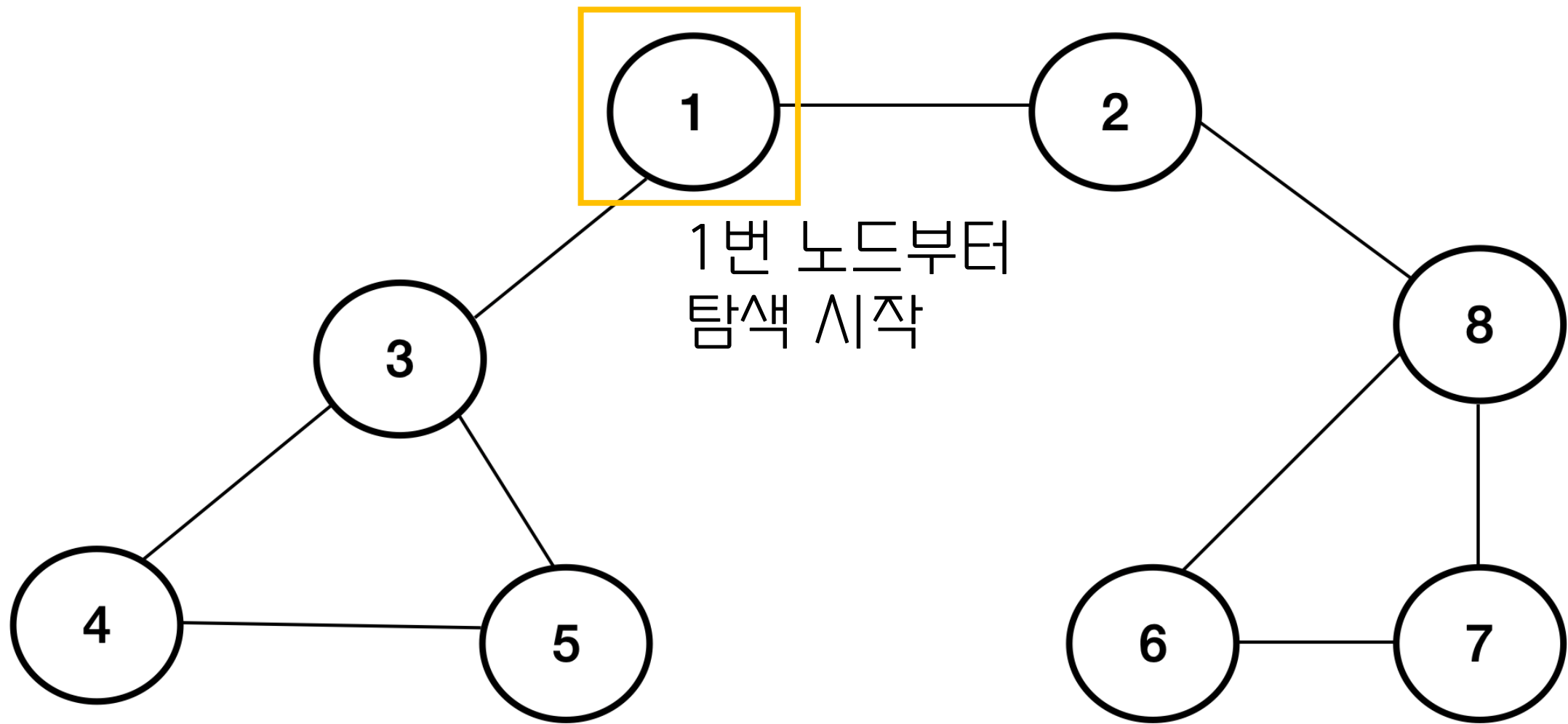
현재 방문 노드 : 1-2-3-8-4-5-6-7

너비우선탐색 Depth First Search

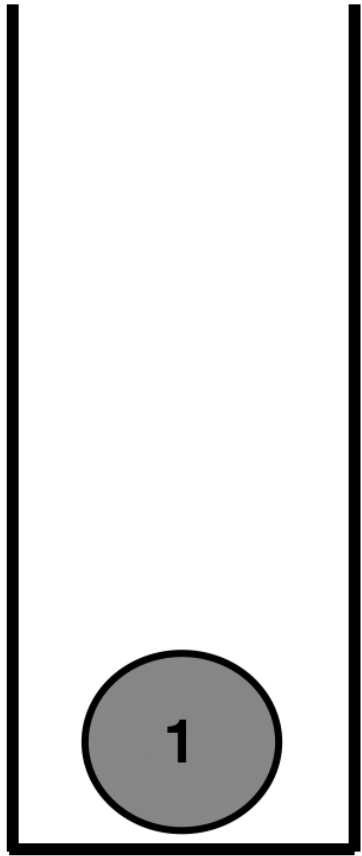
하나의 branch(분기)를 끝까지 탐색하고 다음 분기로 넘어가는 탐색법
= 깊게 탐색하는 방법
(미로를 한 방향으로 끝까지 갔다가 돌아오는 것과 유사함)

후입선출, **Stack(or 재귀)**을 이용하여 구현한다.

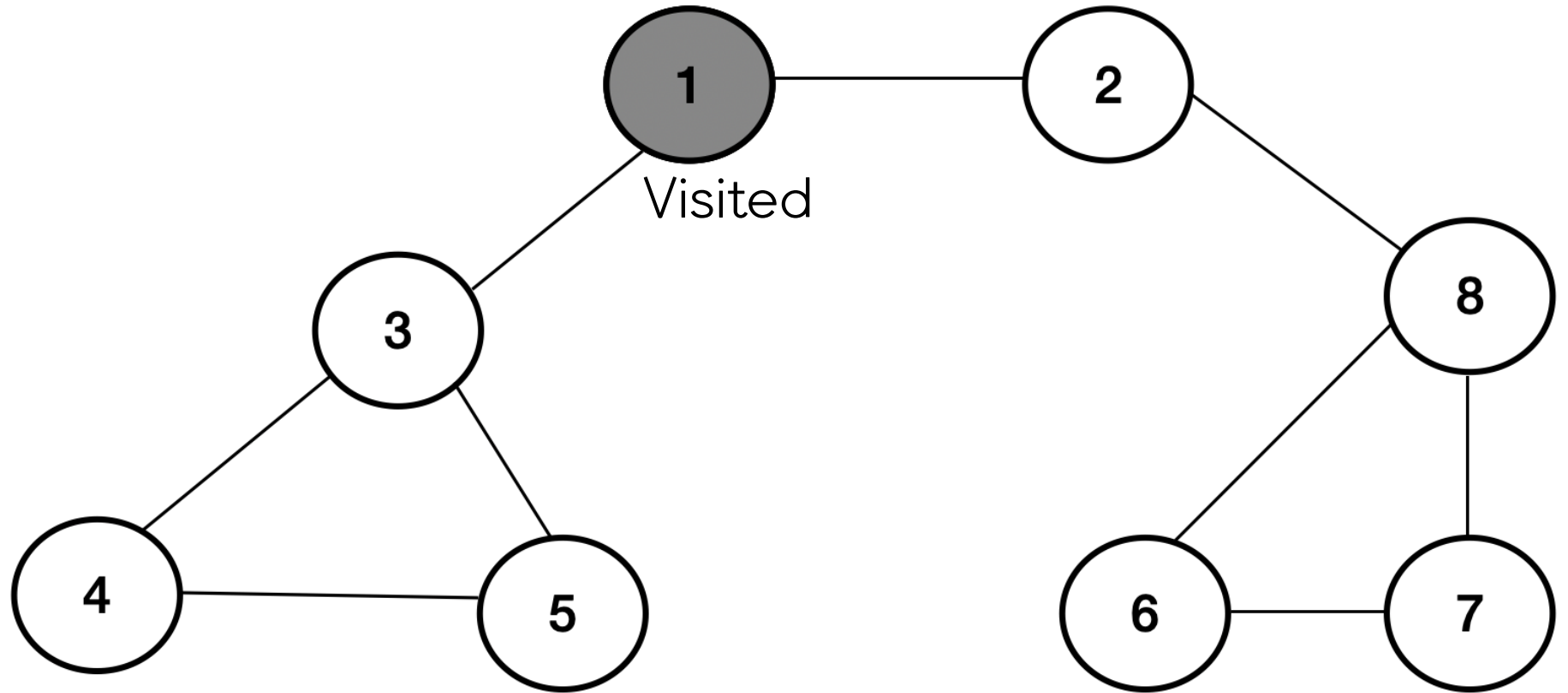
1. 스택에서 노드 A를 꺼냄
2. 노드A 에 인접한 노드 B,C,... 를 스택에 삽입
3. 1~2 반복.



루트노드를 스택에 삽입, 방문처리



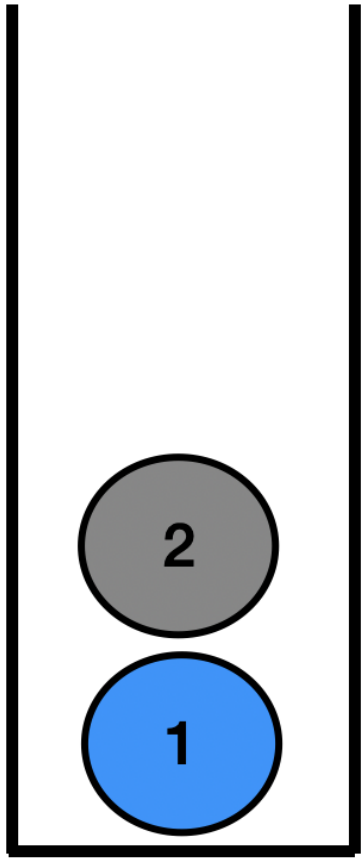
스택



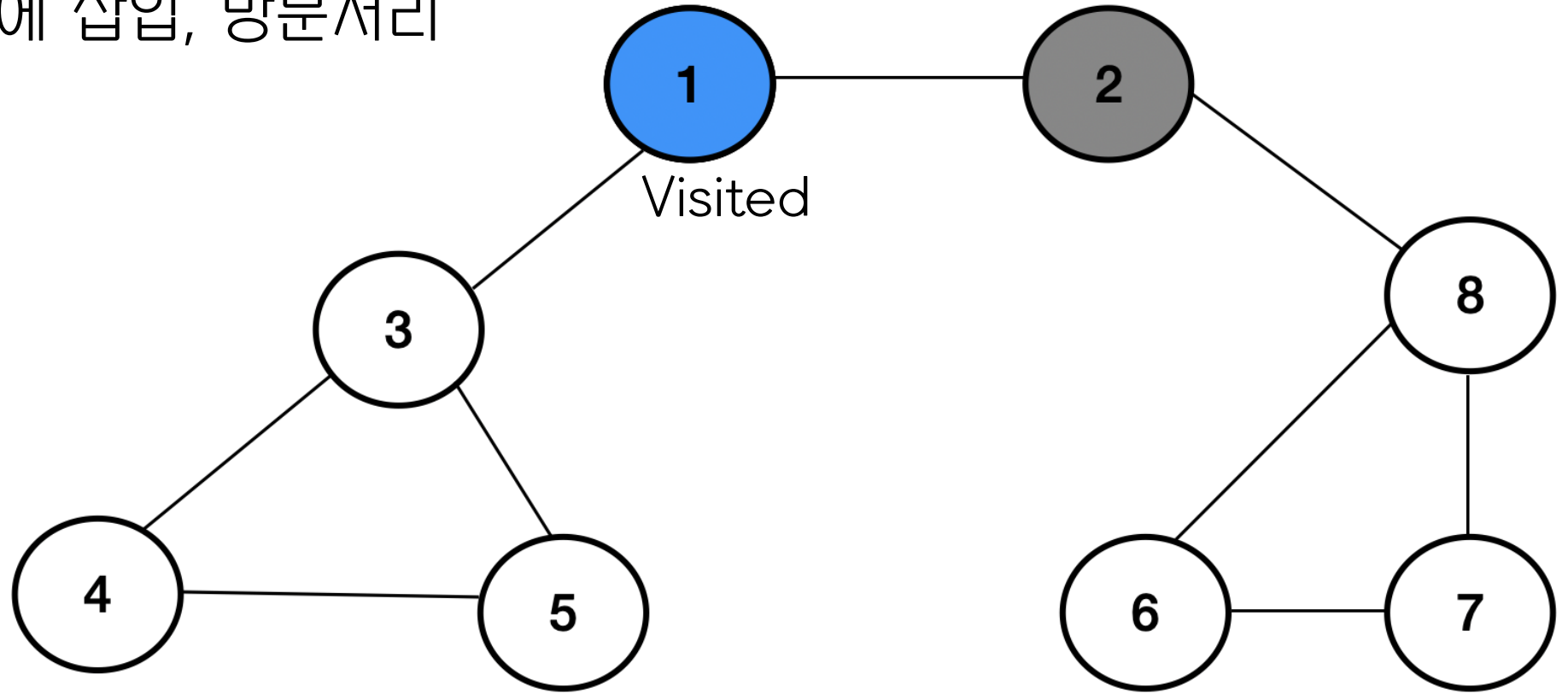
<https://heytech.tistory.com/>

현재 방문 노드 : 1

스택에서 노드 개념
인접노드 중 번호가 낮은 노드1개를
스택에 삽입, 방문처리



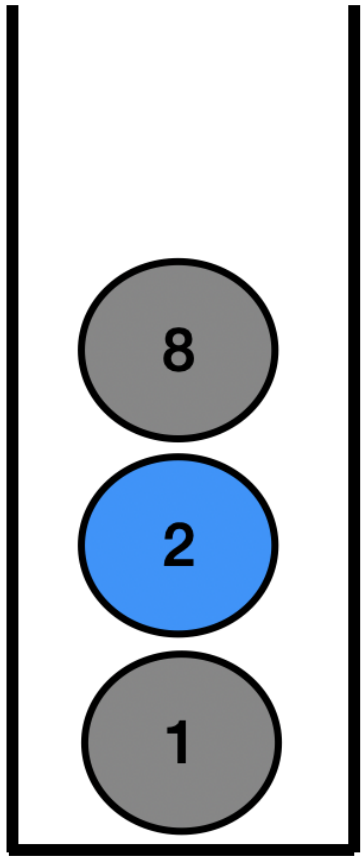
스택



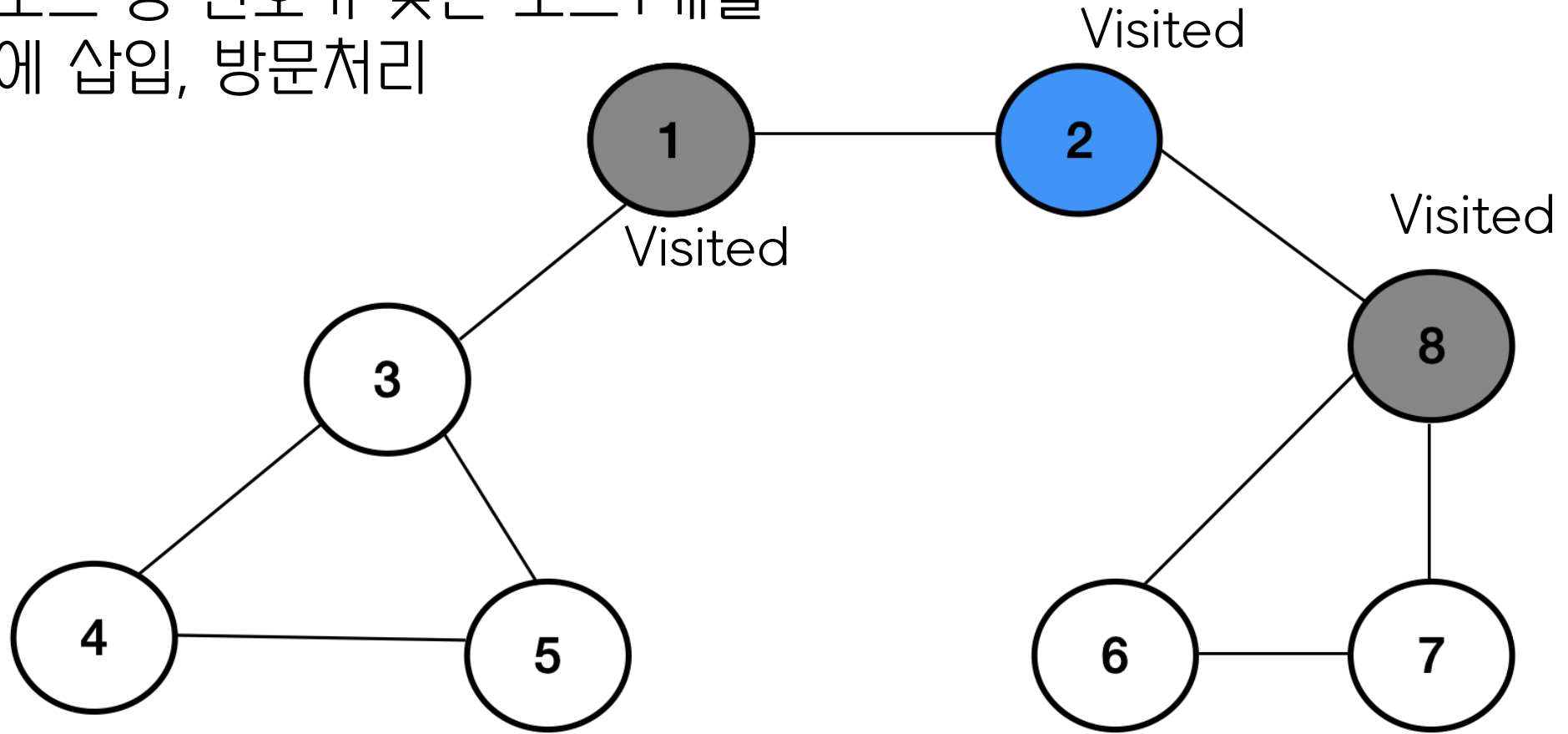
<https://heytech.tistory.com/>

현재 방문 노드 : 1-2

스택에서 노드 개념
인접노드 중 번호가 낮은 노드1개를
스택에 삽입, 방문처리



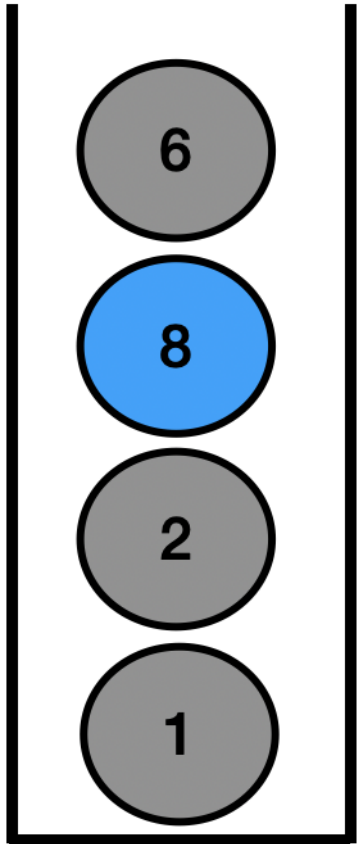
스택



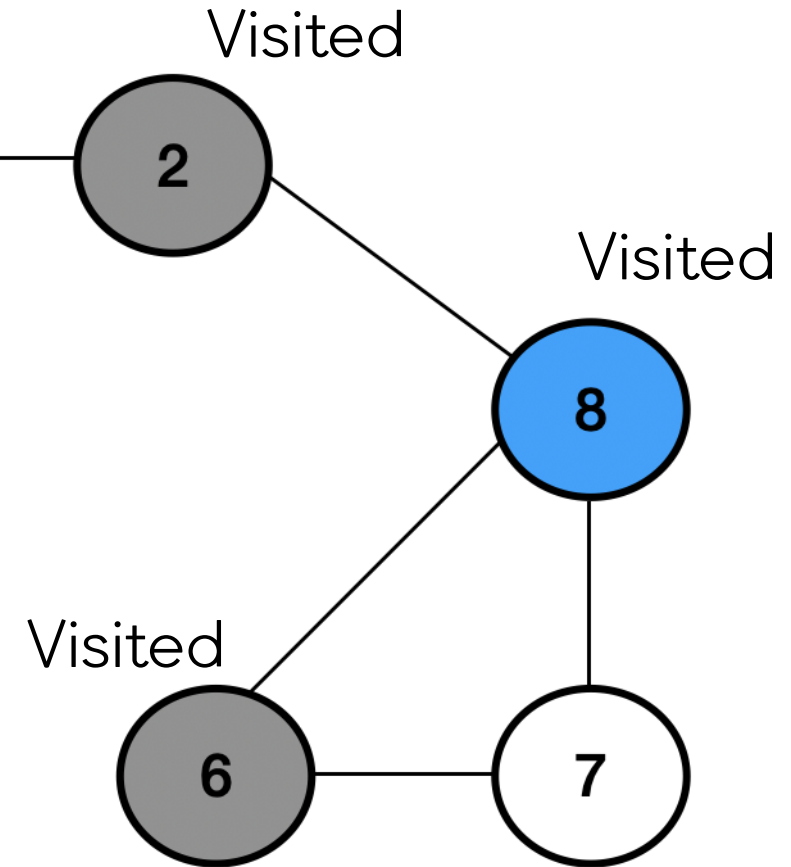
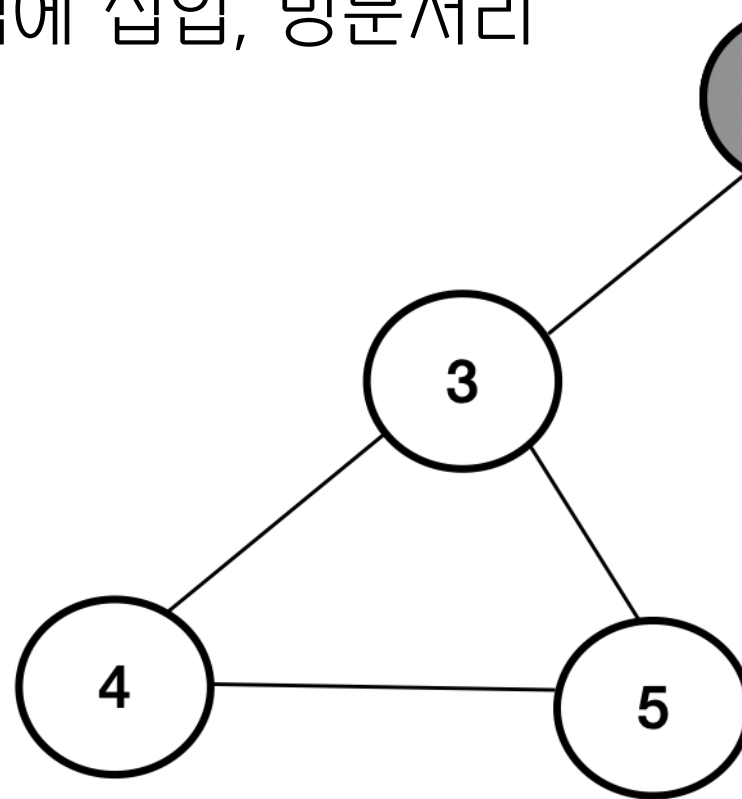
<https://heytech.tistory.com/>

현재 방문 노드 : 1-2-8

스택에서 노드 개념
인접노드 중 번호가 낮은 노드1개를
스택에 삽입, 방문처리



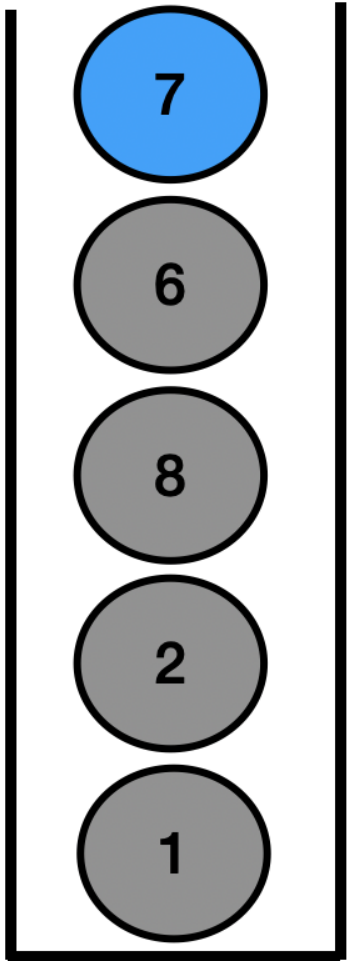
스택



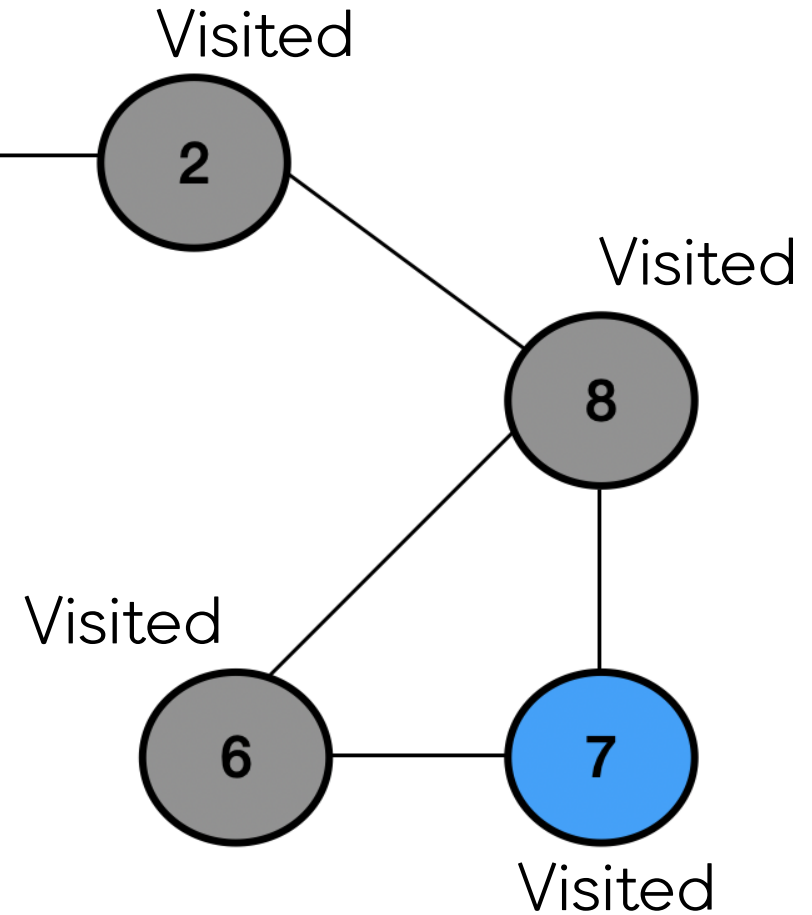
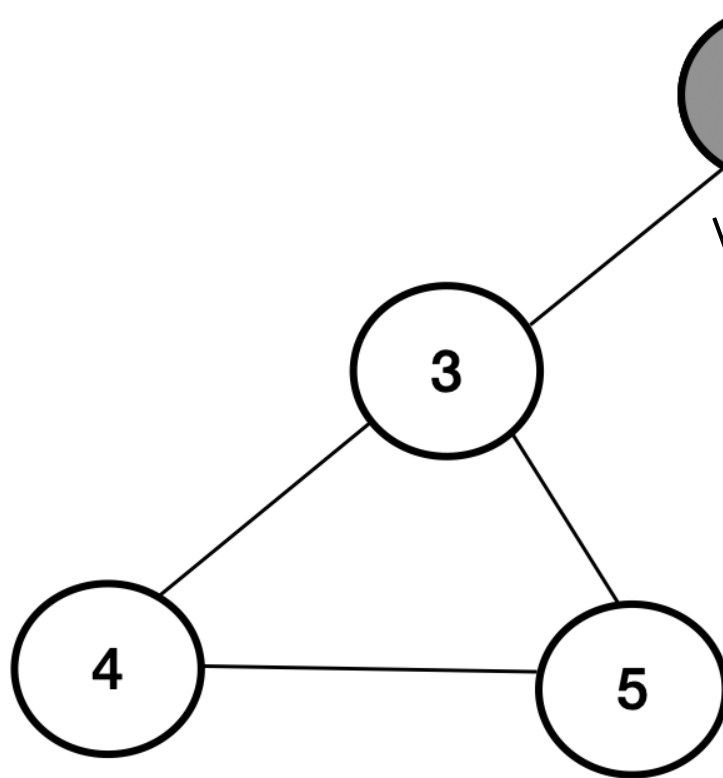
<https://heytech.tistory.com/>

현재 방문 노드 : 1-2-8-6

스택에서 노드 개념
인접노드 중 번호가 낮은 노드1개를
스택에 삽입, 방문처리



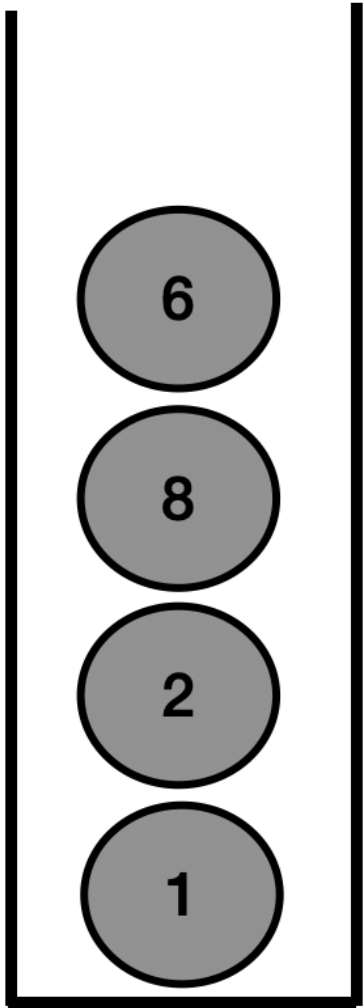
스택



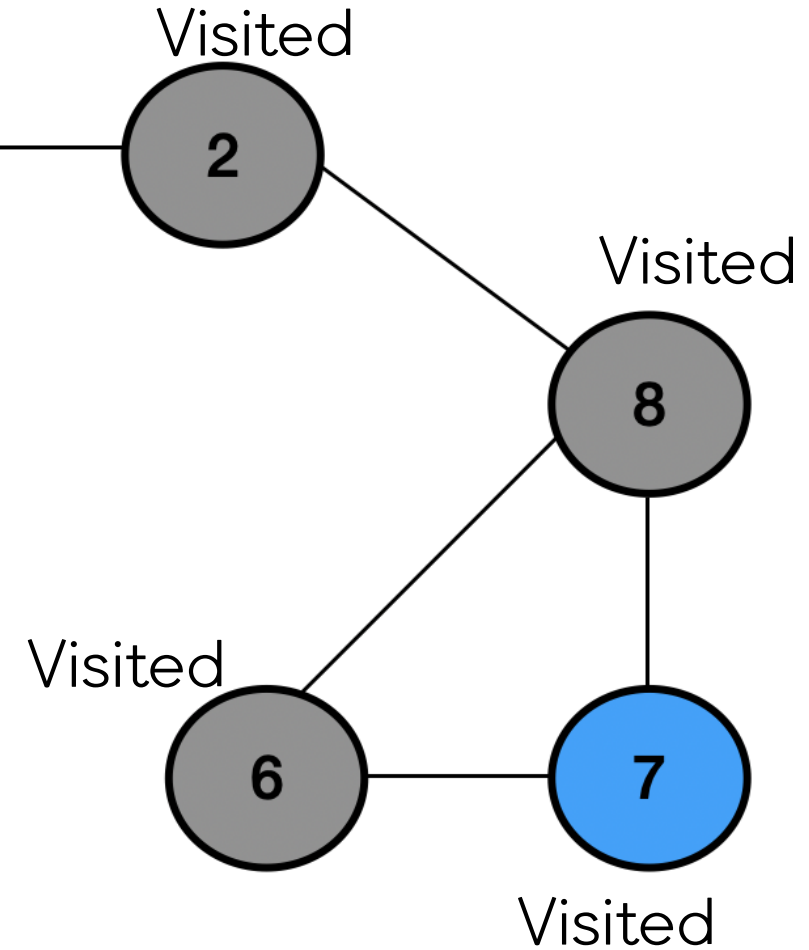
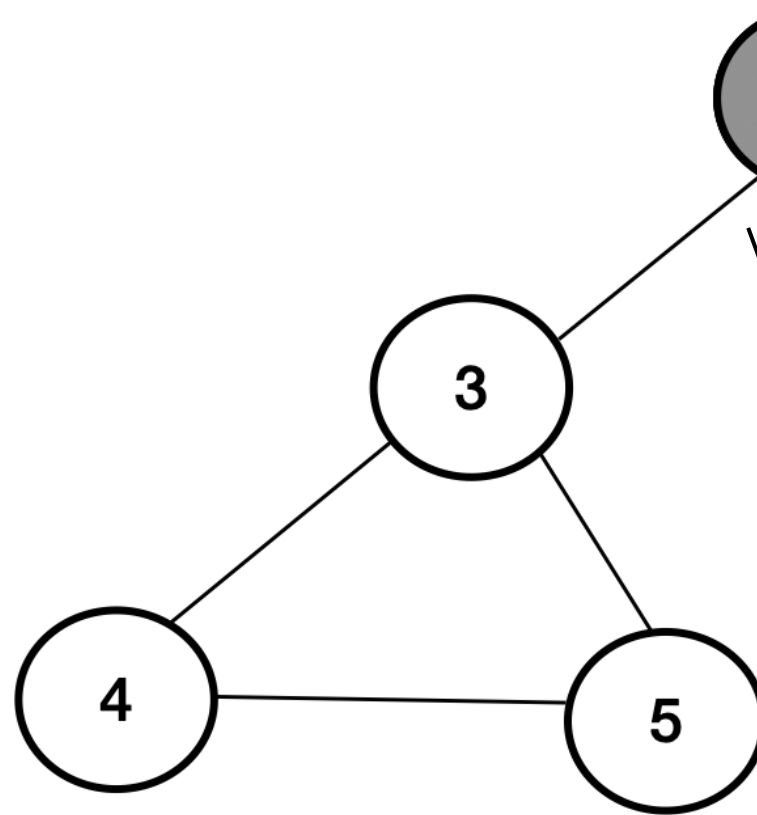
<https://heytech.tistory.com/>

현재 방문 노드 : 1-2-8-6-7

스택에서 노드 개념
인접노드 중 번호가 낮은 노드1개를
스택에 삽입, 방문처리 (7번 -> 없음)



스택

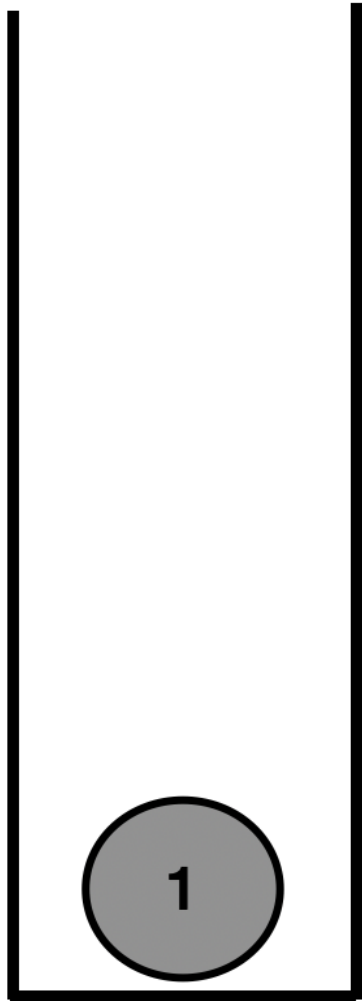


현재 방문 노드 : 1-2-8-6-7

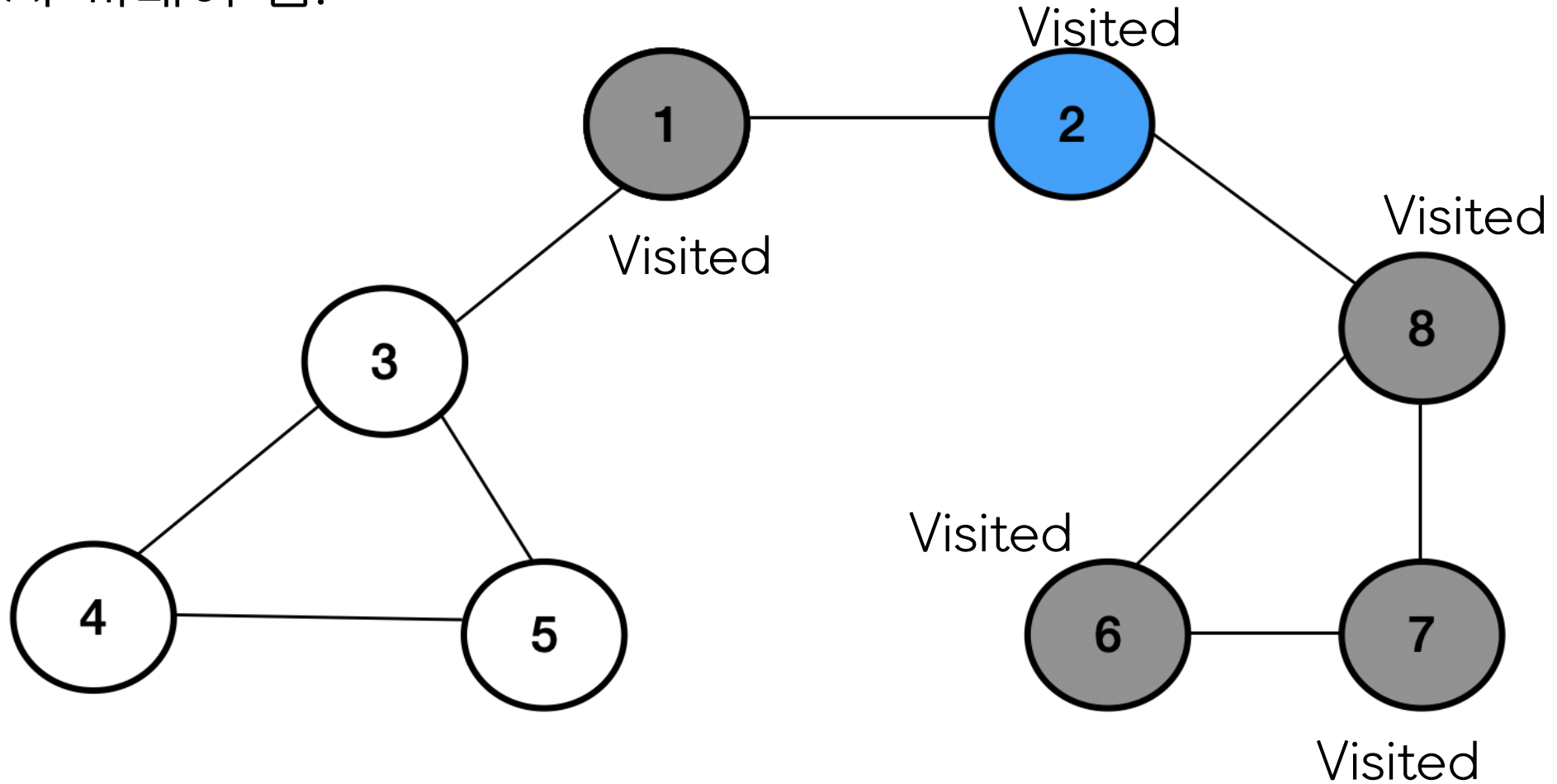
스택에서 노드 꺼냄

6, 8, 2에 대해서 인접하고 미방문한 노드가 없음

1까지 꺼내야 함.

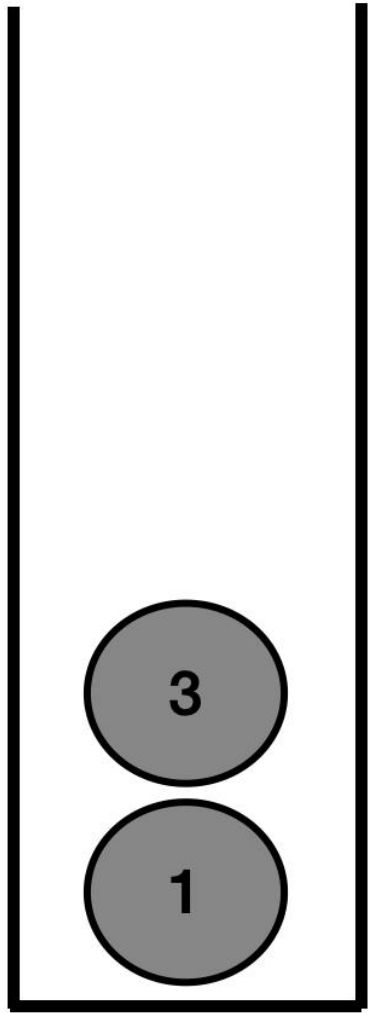


스택

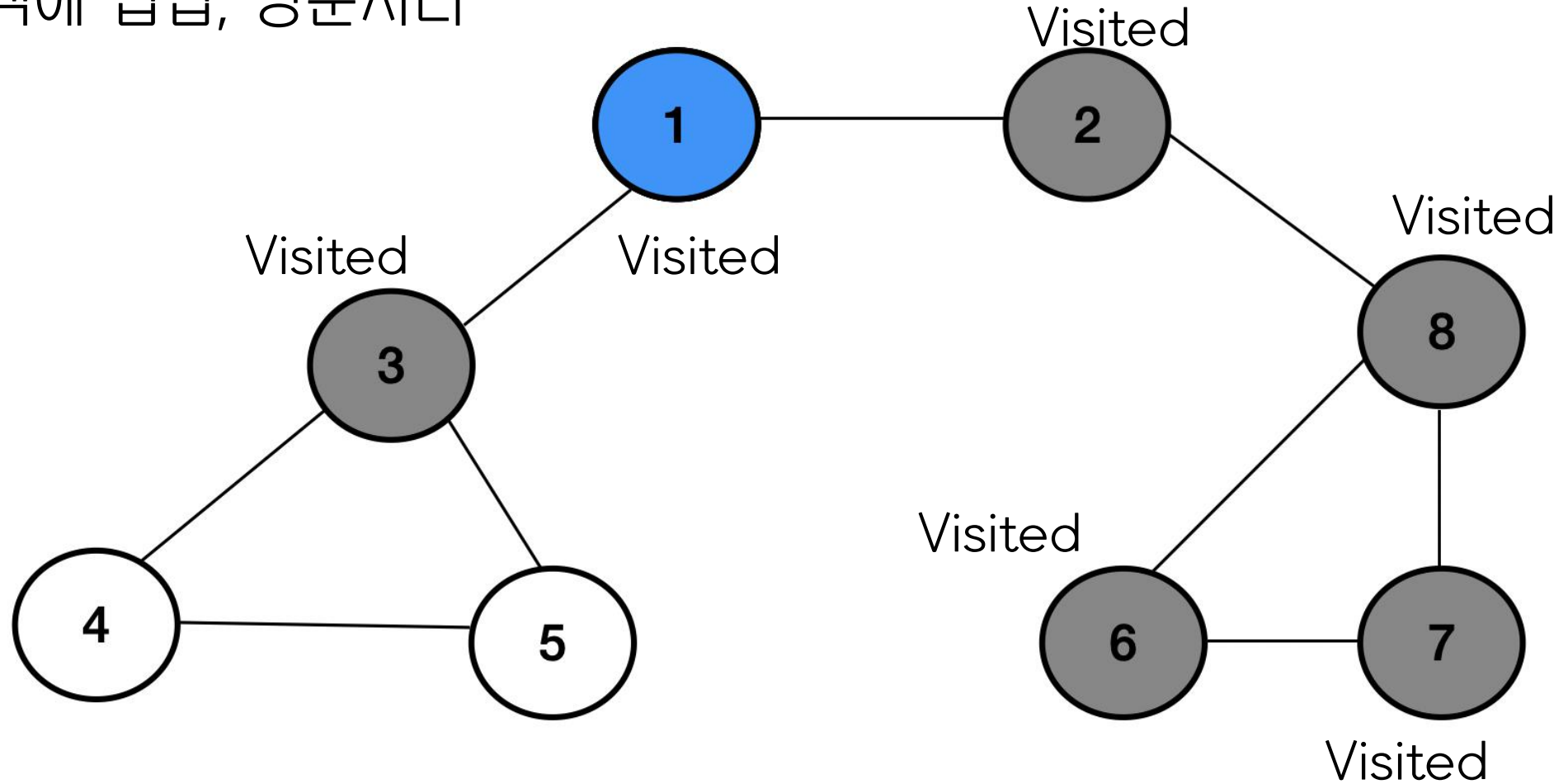


현재 방문 노드 : 1-2-8-6-7 <https://heytech.tistory.com/>

스택에서 노드 개념
인접노드 중 번호가 낮은 노드1개를
스택에 삽입, 방문처리

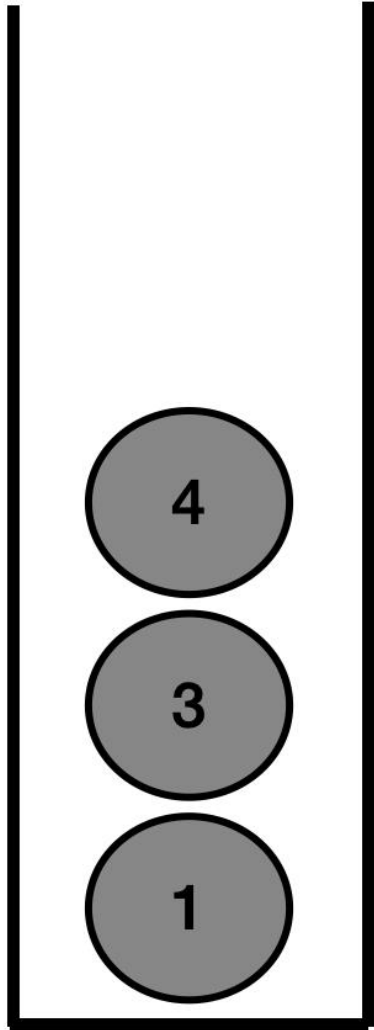


스택

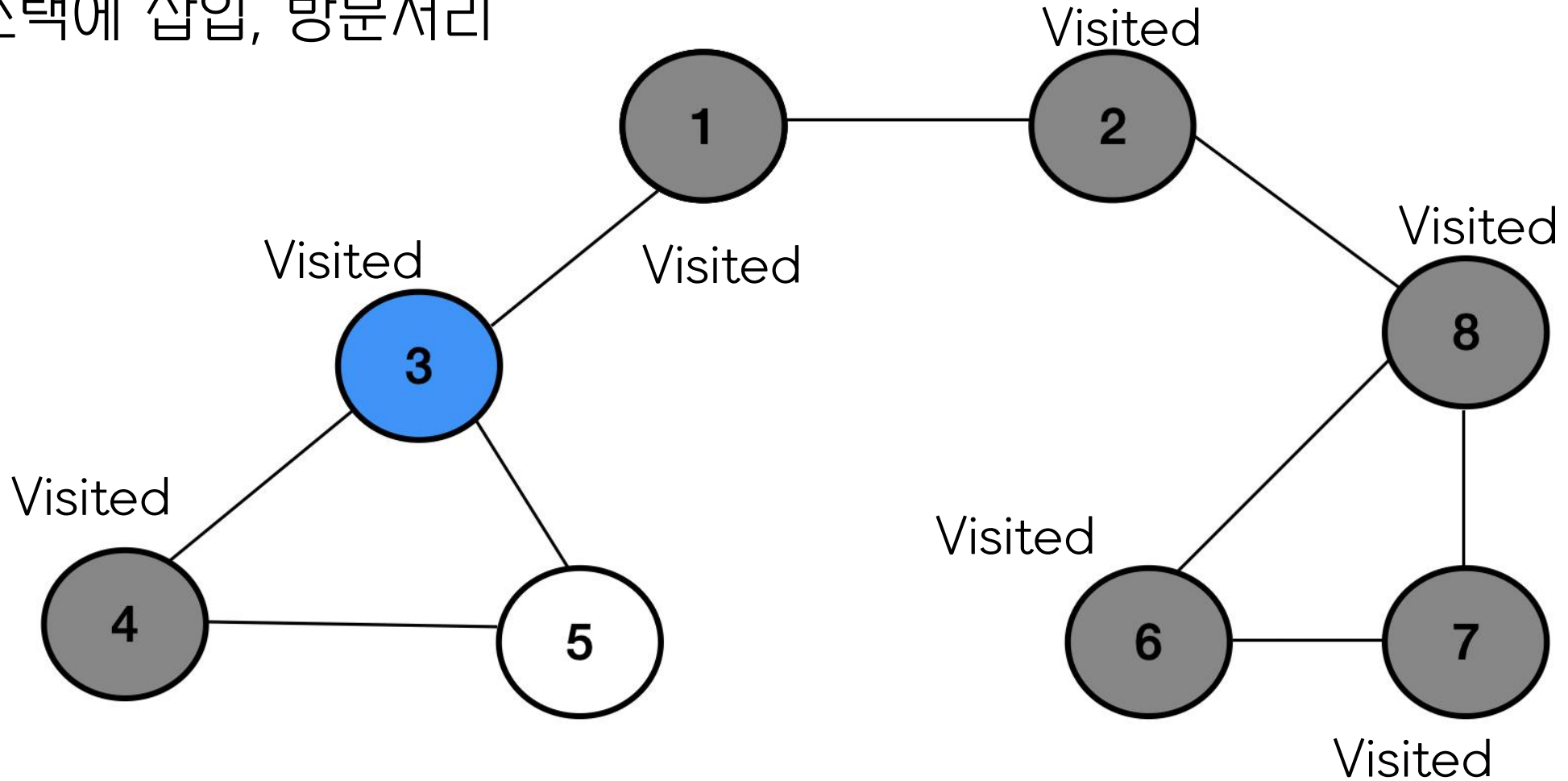


현재 방문 노드 : 1-2-8-6-7-3 <https://heytech.tistory.com/>

스택에서 노드 개념
인접노드 중 번호가 낮은 노드1개를
스택에 삽입, 방문처리

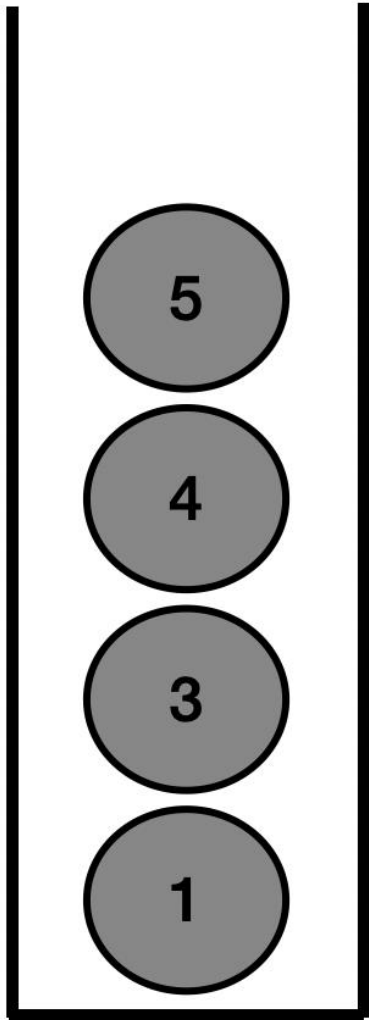


스택

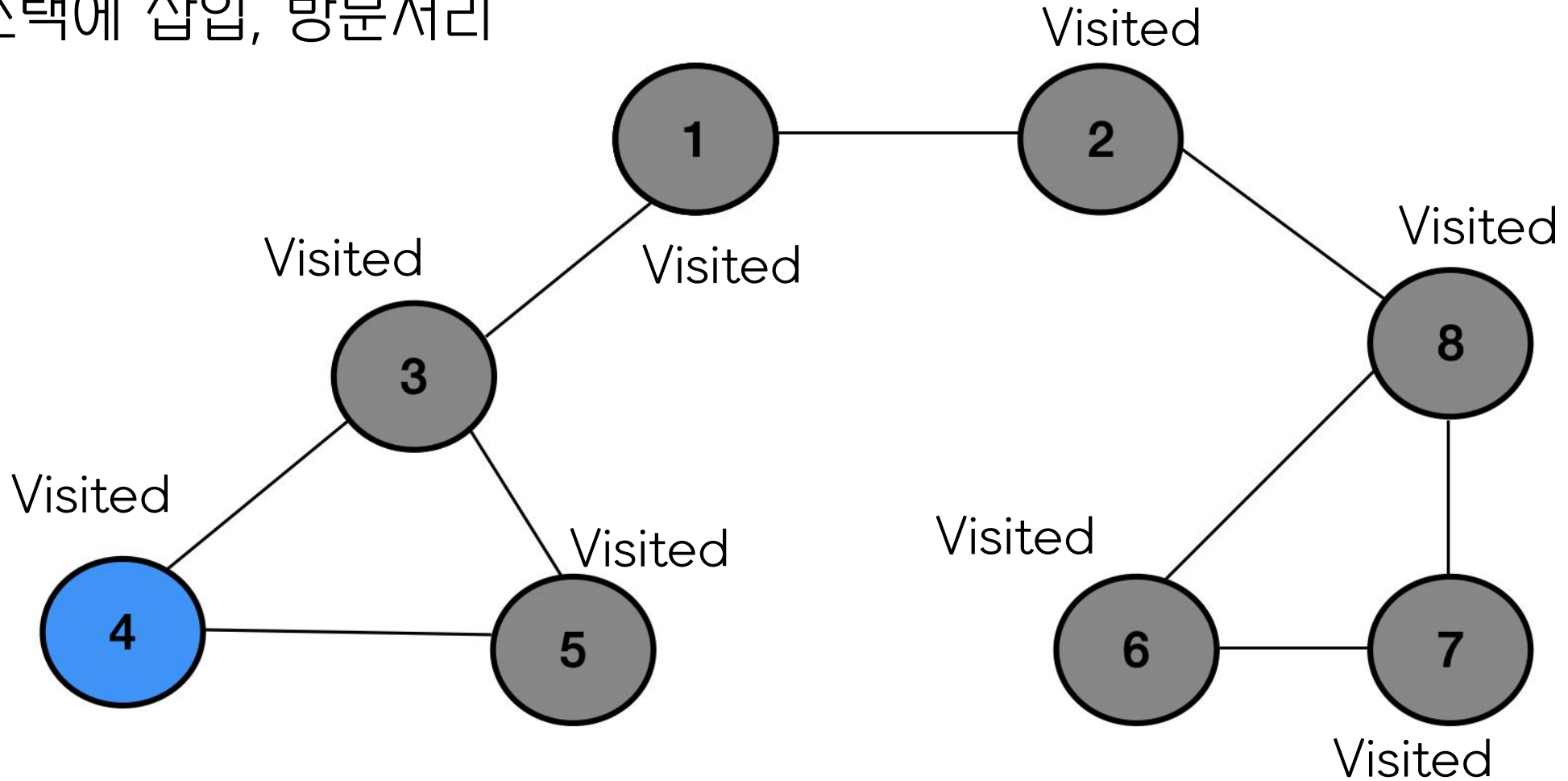


현재 방문 노드 : 1-2-8-6-7-3-4 <https://heytech.tistory.com/>

스택에서 노드 개념
인접노드 중 번호가 낮은 노드1개를
스택에 삽입, 방문처리



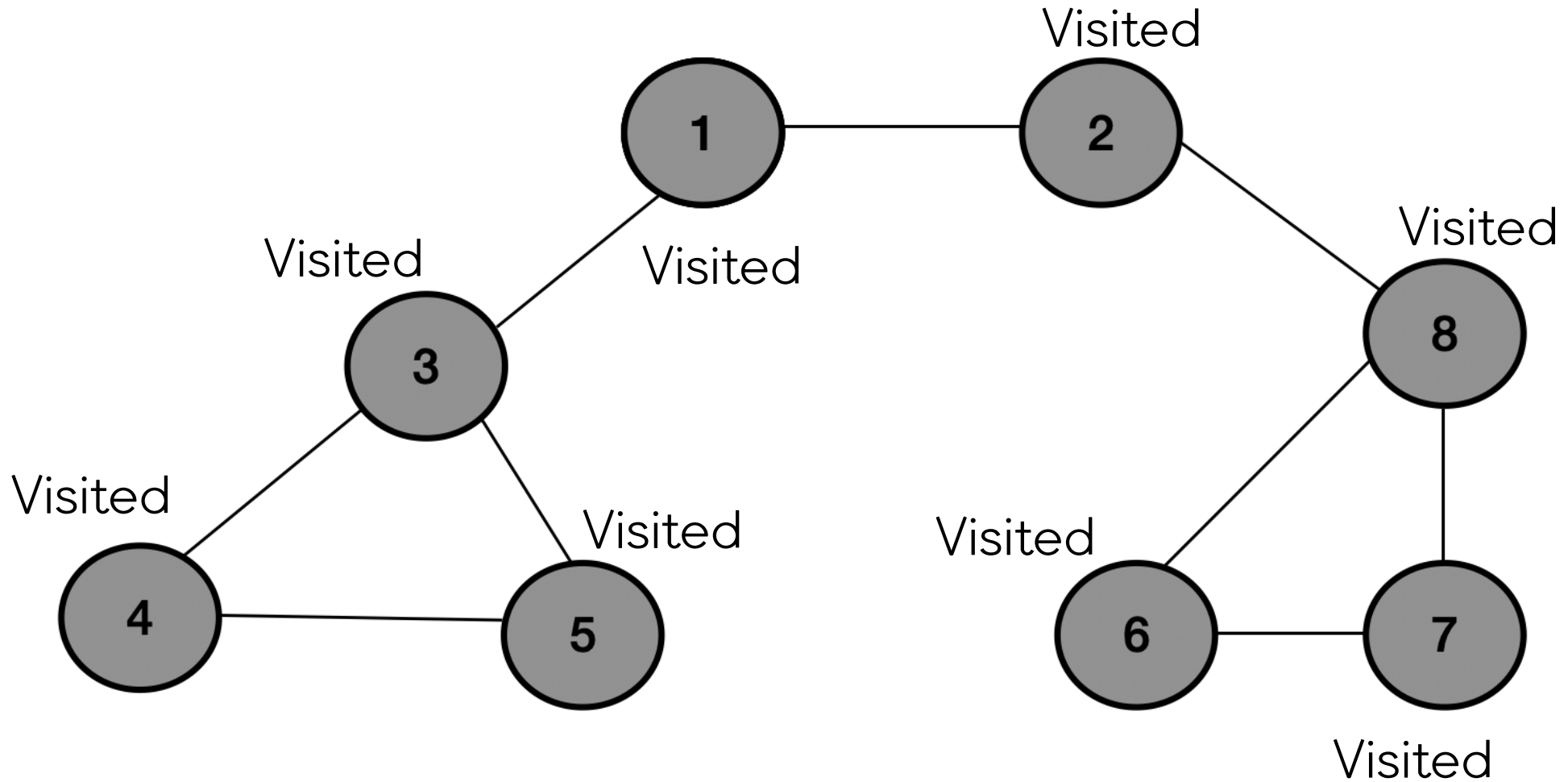
스택



현재 방문 노드 : 1-2-8-6-7-3-4-5

<https://heytechtistory.com/>

스택 빌 때까지 노드개념
스택 비면 탐색 종료!



스택

현재 방문 노드 : 1-2-8-6-7-3-4-5

<https://heytech.tistory.com/>

재귀호출로 구현하는 DFS

재귀호출 자체가 후입선출 구조이므로
재귀적인 함수 호출 만으로도
DFS 구현이 가능하다.

즉, 스택을 구현 및 사용할 필요가 없음.

운영체제 자체가 스택의 역할

코드로 보는 BFS 와 DFS

[Senior/week5/BFSandDFS.py](#)

[Senior/week5/DFS_recur.py](#)

2차원 배열을 탐색하는 문제

N×M크기의 배열로 표현되는 미로가 있다.

1	0	1	1	1	1
1	0	1	0	1	0
1	0	1	0	1	1
1	1	1	0	1	1

[Senior/week5/mazeSearch.py](#)

벽이 있는 형태의 2차원 배열을 입력받고
미로 탈출, 감염진행 등등...

탐색 방법 :

배열을 그래프라고 인식하자.

방문 가능한 칸들을 노드라고 할 수 있다.

인접하고, 방문하지 않았고, 접근가능한 노드로 이동하는 BFS,DFS 탐색을 적용하자.

그래프 탐색 문제 유형

1. (N, M) 으로 이동할 수 있나?
2. (N, M) 으로 가는데 몇 칸을 거쳐야 하는가?
3. 몇 개의 부분으로 나눌 수 있는가?
4. 트리에 대한 탐색

DFS와 BFS로 풀어보는 문제들

〈Basic〉

1260 BFS와 DFS

<https://www.acmicpc.net/problem/1260>

2606 바이러스

<https://www.acmicpc.net/problem/2606>

1012 유기농 배추

<https://www.acmicpc.net/problem/1012>

〈Lv Up〉

1389 케빈 베이컨의 6단계 법칙

<https://www.acmicpc.net/problem/1389>

7569 토마토

<https://www.acmicpc.net/problem/7569>

2206 벽 부수고 이동하기

<https://www.acmicpc.net/problem/2206>

〈Hard〉

14502 연구소

<https://www.acmicpc.net/problem/14502>

15591 MooTube

<https://www.acmicpc.net/problem/15591>

9019 DSLR

<https://www.acmicpc.net/problem/9019>

python3 제출 안되면

pypy3으로 시도

2학기 씨앗에서는

1. 비기너-시니어 통합, 활동 6번쯤 예상.
더 많은 학우들에게 도움이 될 수 있게 알찬 내용으로 돌아오겠습니다.
2. 교내 프로그래밍 대회 이후 문제 리뷰
2주정도 진행할 예정
3. 새로운 이벤트 계획중
취업한 선배님과 뭔가 이벤트 기획중