DFS/BFS/RECURSION

<search>

기본 개념

- stack, queue, recursion

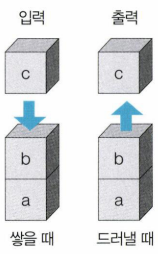
stack과 queue의 핵심 함수 2개

- push

- pop

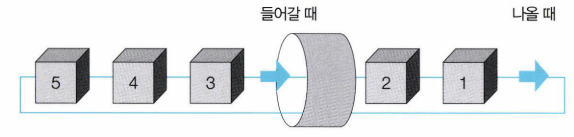
stack?

- 박스 쌓기와 같음. 선입후출(FILO) or 후입선출(LIFO).S



queue?

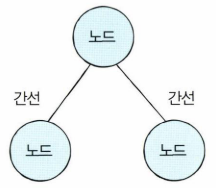
* 놀이공원 대기줄과 같음. 선입선출(FIFO).



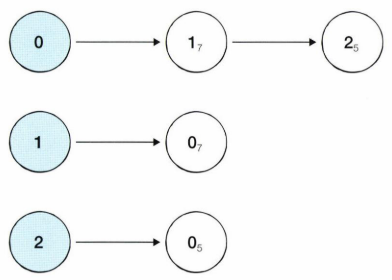
recursion?

* 재귀 함수의 동작 원리는 스택 자료구조와 동일. ex) DFS
* 반복문보다 코드가 간결함.

DFS(깊이 우선 탐색)?

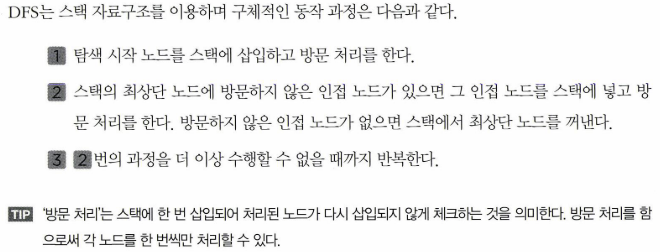
이 때 노드를 정점(vertex)라고도 한다.

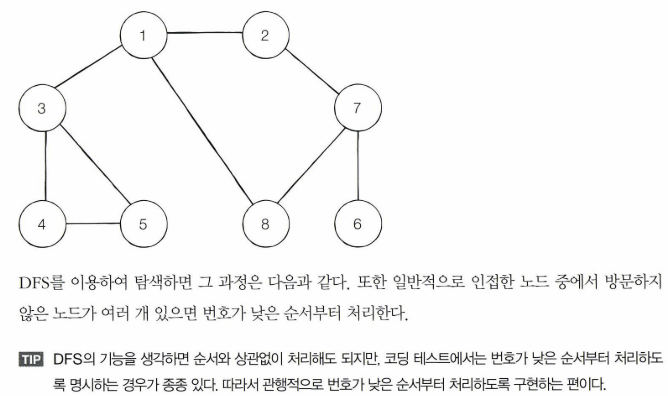


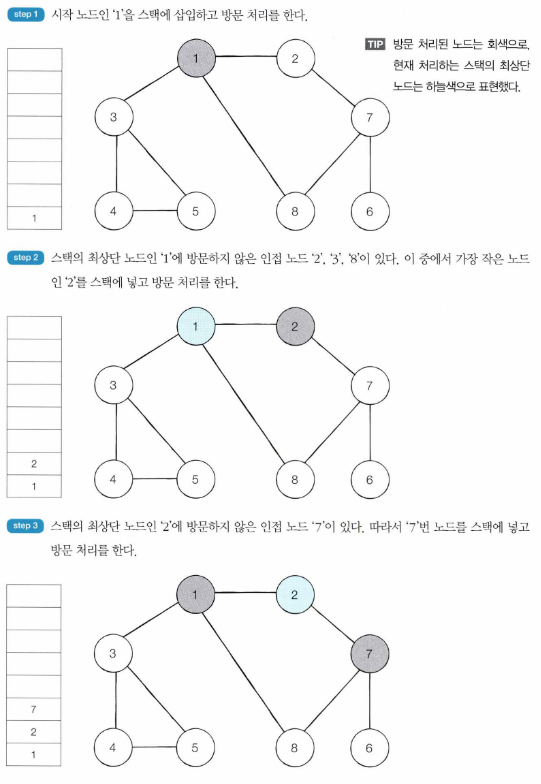


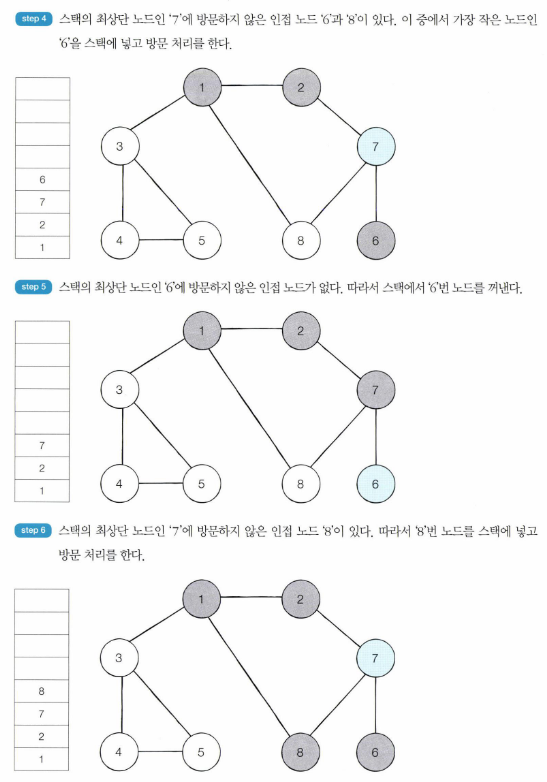
* 인접 행렬 방식: 모든 관계를 저장하므로 노드 개수가 많을수록 메모리가 불필요하게 낭비됨.
* 인접 리스트 방식: 연결된 정보만을 저장하므로 메모리를 효율적으로 사용 가능. 연결된 데이터를 하나씩 확인해야 해서 특정 두 노드가 연결되었는지 정보를 얻는 속도가 인접 행렬 방식에 비해 느림.

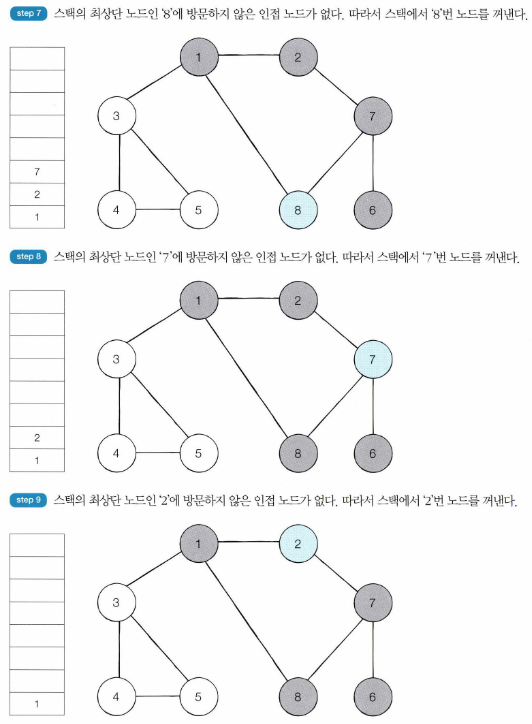
**DFS**

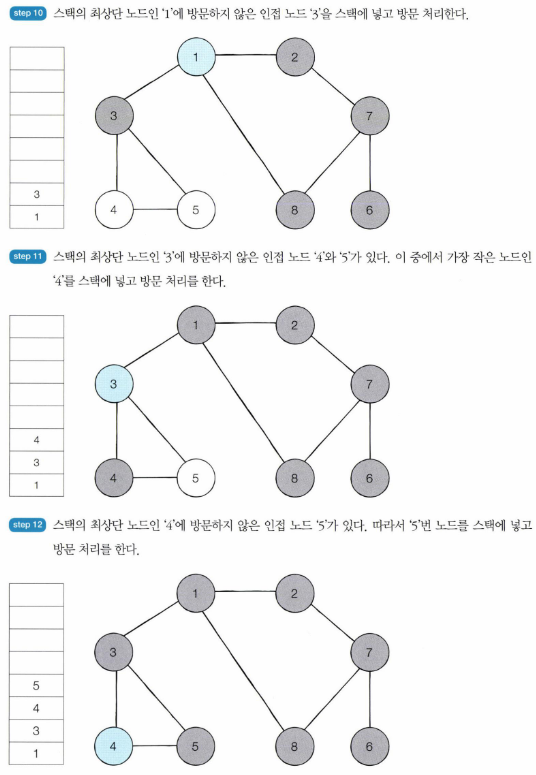


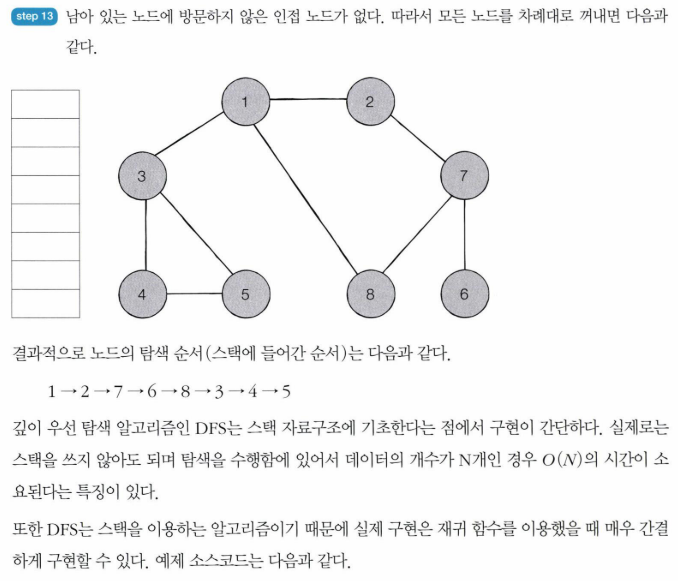




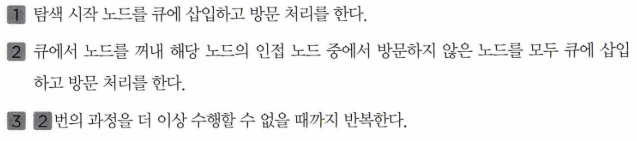


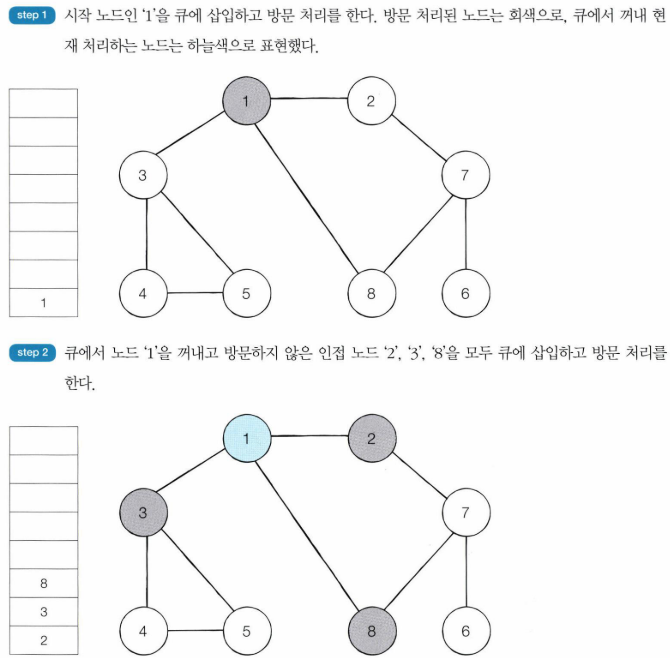


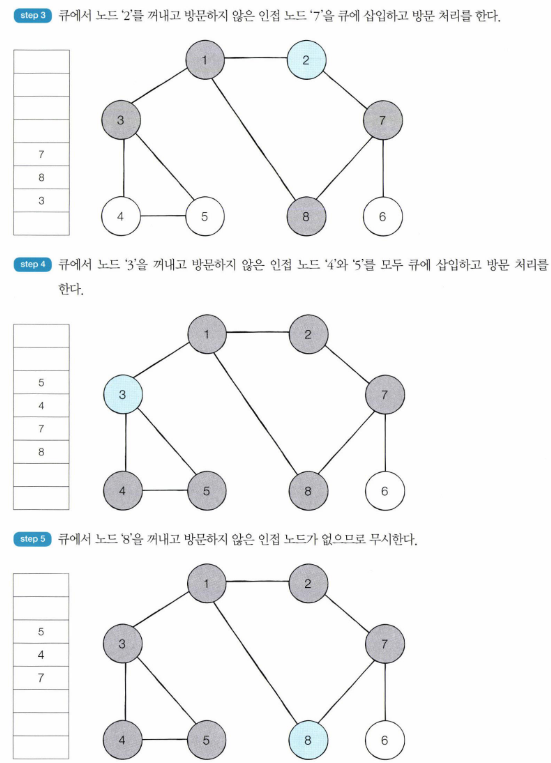


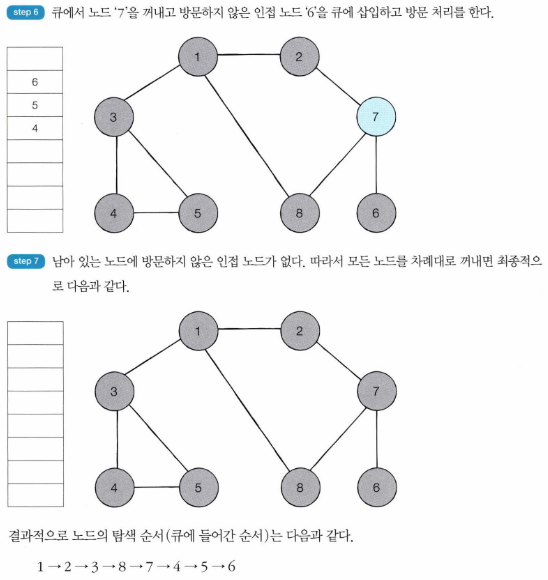


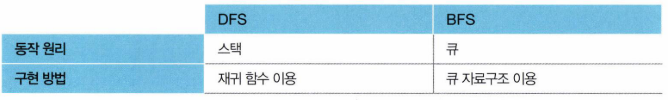
**BFS**

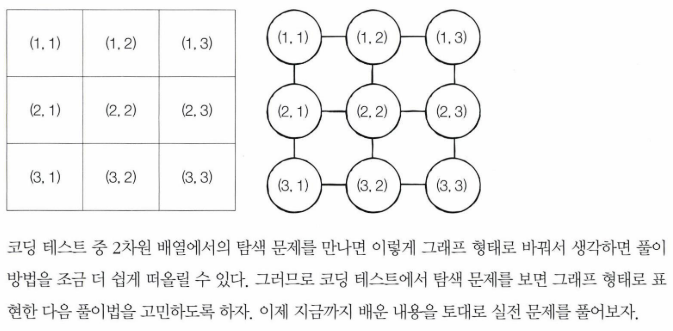
****

****

****

****

****

****