

네트워크 토폴리지와 병목 현상

🕒 생성일	@2023년 12월 2일 오후 11:28
🏷 태그	네트워크

목차

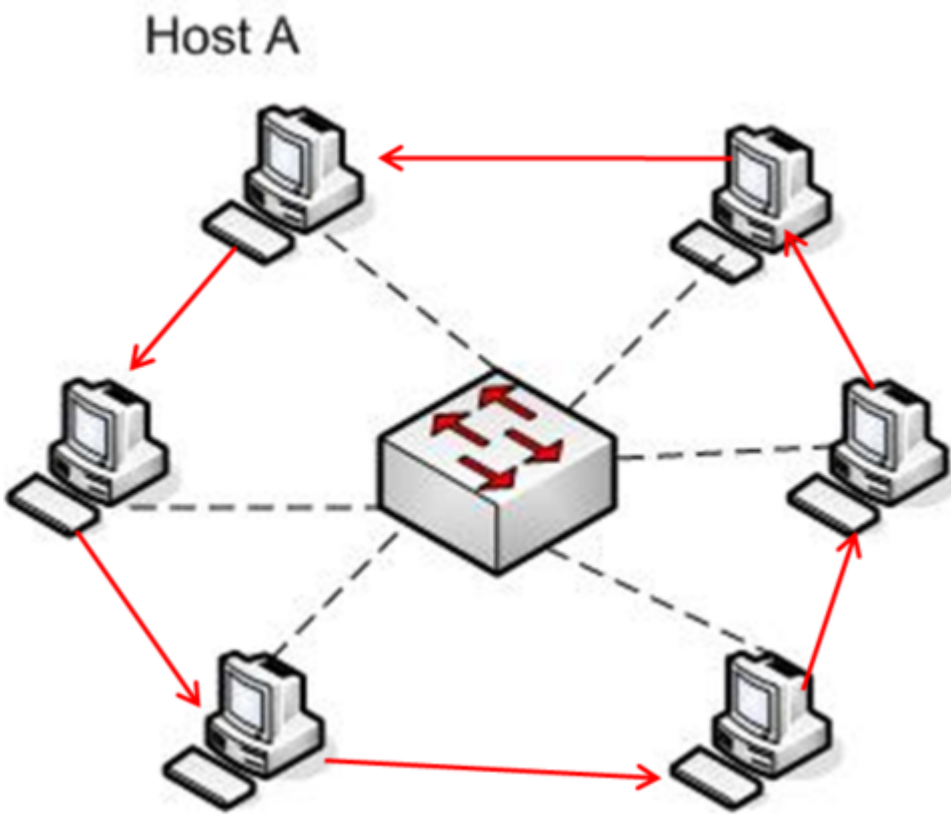
- 네트워크 토폴리지
 - 네트워크 토폴리지란?
 - 트리 토폴리지
 - 버스 토폴리지
 - 스타(star, 성형) 토폴리지
 - 링형 토폴리지
 - 메시 토폴리지
- 병목 현상
 - 병목현상이란?
 - 예시

네트워크 토폴리지

네트워크 토폴리지란?

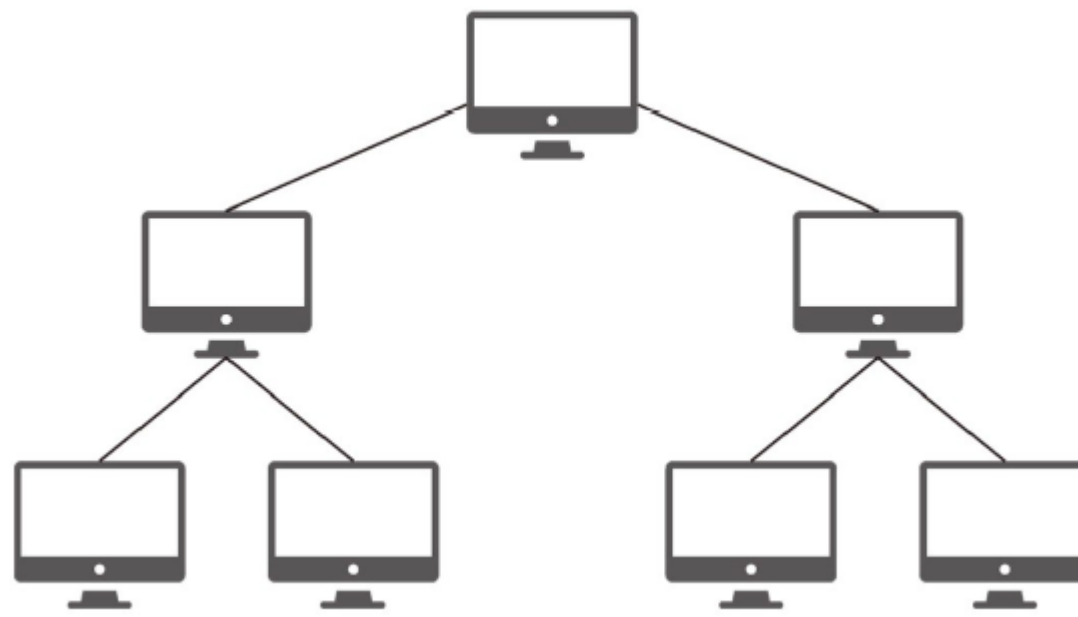
- 네트워크(Network)란 상호 간에 정보를 교환할 수 있도록 유선, 무선을 통하여 연결된 형태
- 네트워크 토폴로지는 컴퓨터 네트워크를 구성하는 요소(링크, 노드)들의 배치 형태, 망 구성 방식
- 종류
 - 물리적 토폴리지: 노드, 링크(요소)들의 배치에 의해 결정됨
 - 논리적 토폴리지: 노드들 사이의 데이터 흐름에 따라 결정됨

▼ 예시



위 그림과 같이 네트워크가 물리적으로 연결되어 있고 붉은색 화살표와 같이 데이터가 순차적으로 흐른다면, 물리적 토폴로지는 성형(Start)이고 논리적 토폴로지는 링형(Ring) 이다.

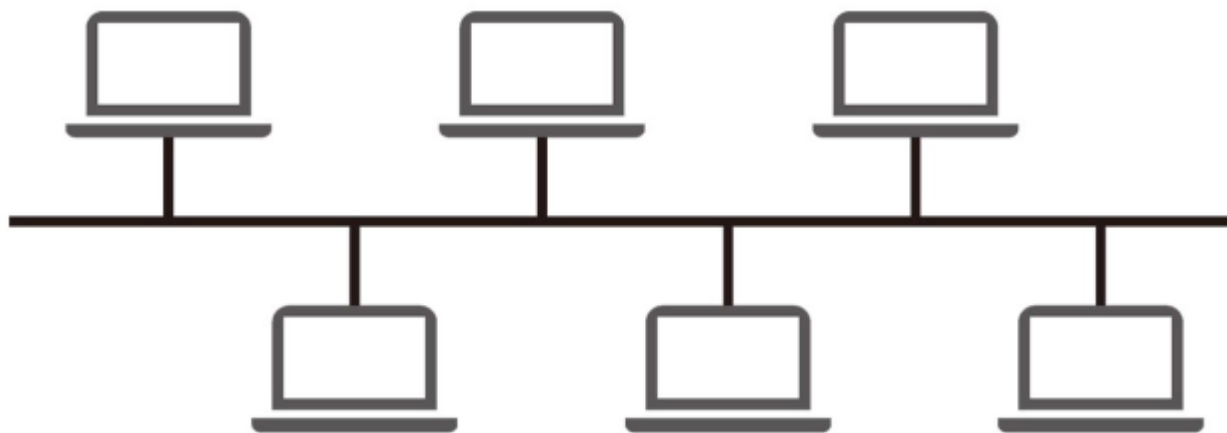
트리 토폴리지



▲ 그림 2-4 트리 토폴로지

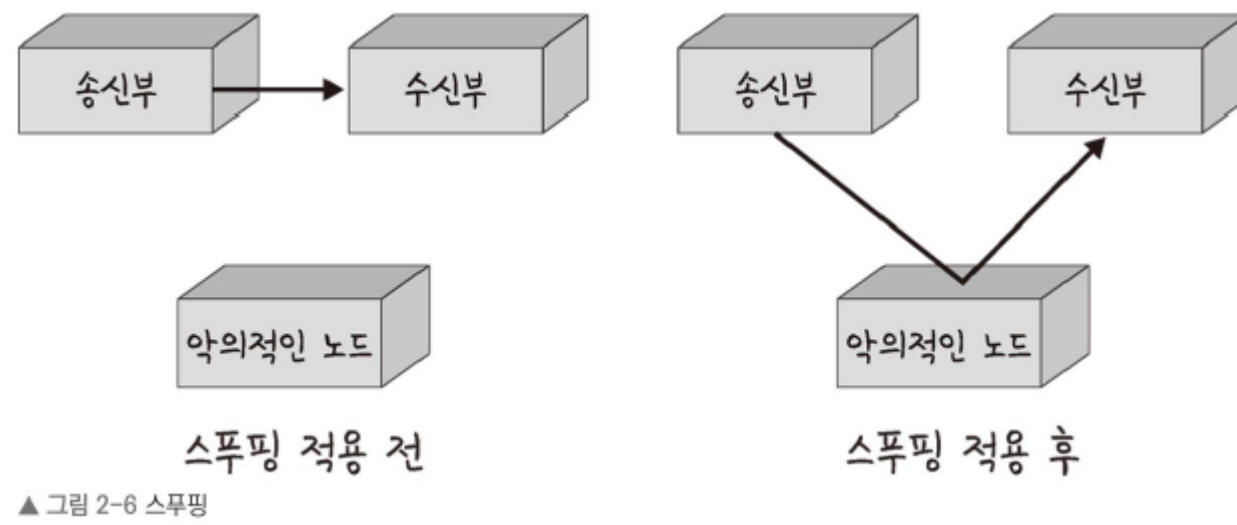
- 계층형 토폴로지라고도 하며 트리 형태로 배치한 네트워크 구성
- 장점
 - 노드의 추가, 삭제가 쉬움
- 단점
 - 특정 노드에 트래픽이 집중될 때 하위 노드에 영향을 끼칠 수 있음

버스 토폴로지

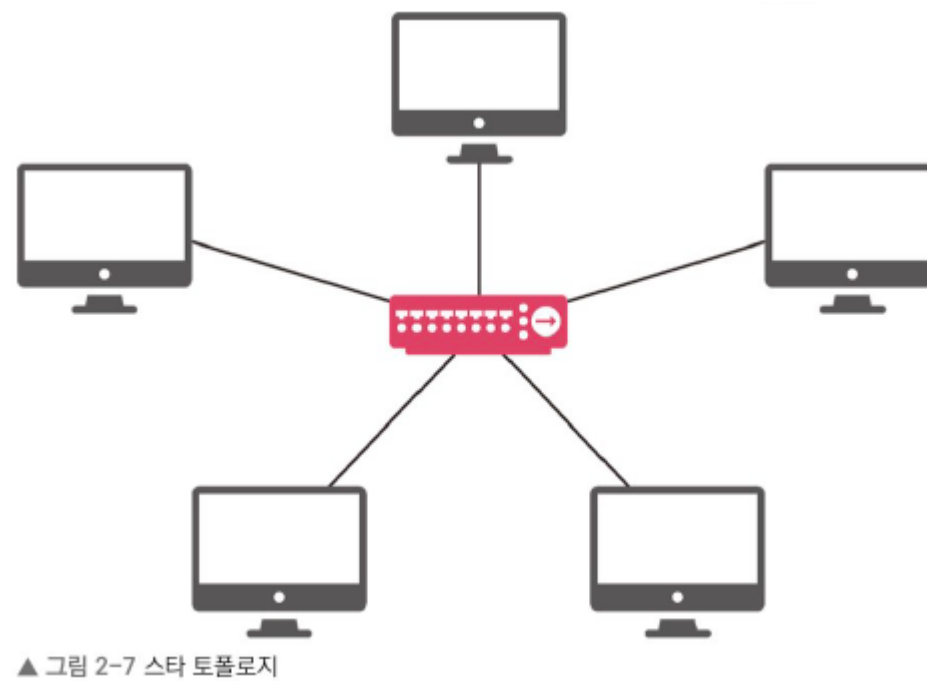


▲ 그림 2-5 버스 토폴로지

- 중앙 통신 회선 하나에 여러 개의 노드가 연결되어 공유하는 네트워크 구성
- 근거리 통신망(LAN)에서 사용
- 장점
 - 설치 비용 ▼ 신뢰성 ▲
 - 중앙 통신 회선에 노드를 추가하거나 삭제하기 쉬움
- 단점
 - 스푸핑 가능
 - ▼ 스푸핑이란?
LAN상에서 송신부의 패킷을 송신과 관련 없는 다른 호스트에 가지 않도록 하는 스위칭 기능을 마비시키거나 속여서 특정 노드에 해당 패킷이 오도록 처리하는 것

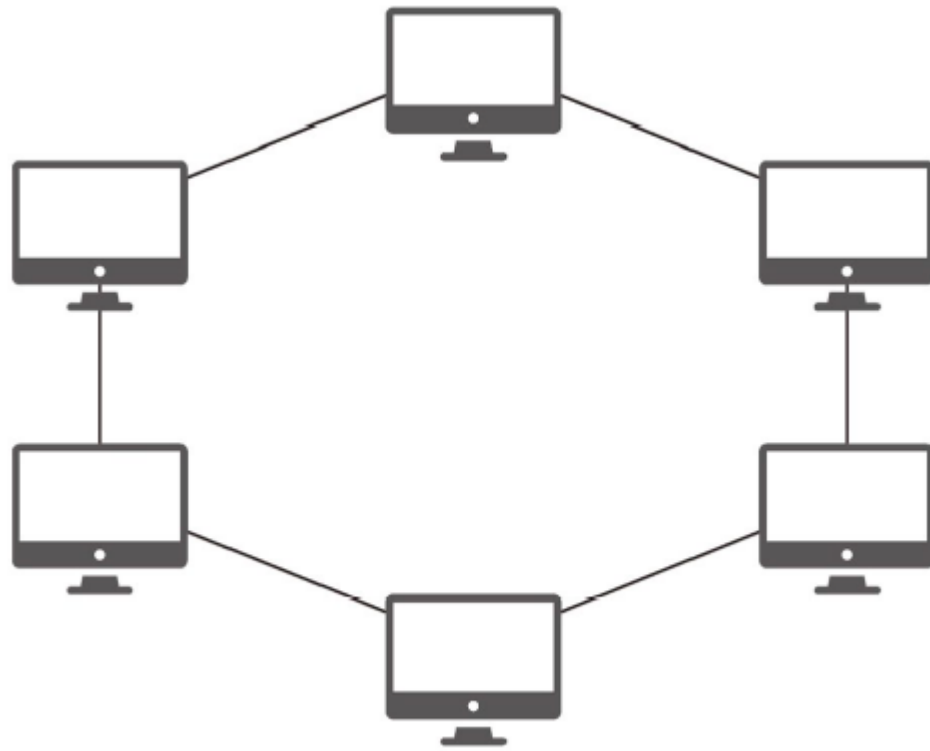


스타(star, 성형) 토폴로지



- 중앙에 있는 노트에 모두 연결된 네트워크 구성
- 장점
 - 노드를 추가하거나 에러를 탐지하기 쉽고 패킷 충돌 가능성 적음
 - 어떤 노드에 장애가 발생해도 쉽게 에러를 발견할 수 있고, 장애 노드가 중앙 노드가 아닐 경우 다른 노드에 영향을 끼치는 것이 적음
- 단점
 - 중앙 노트에 장애 발생 시 전체 네트워크가 사용 불가하고 설치 비용이 고가임

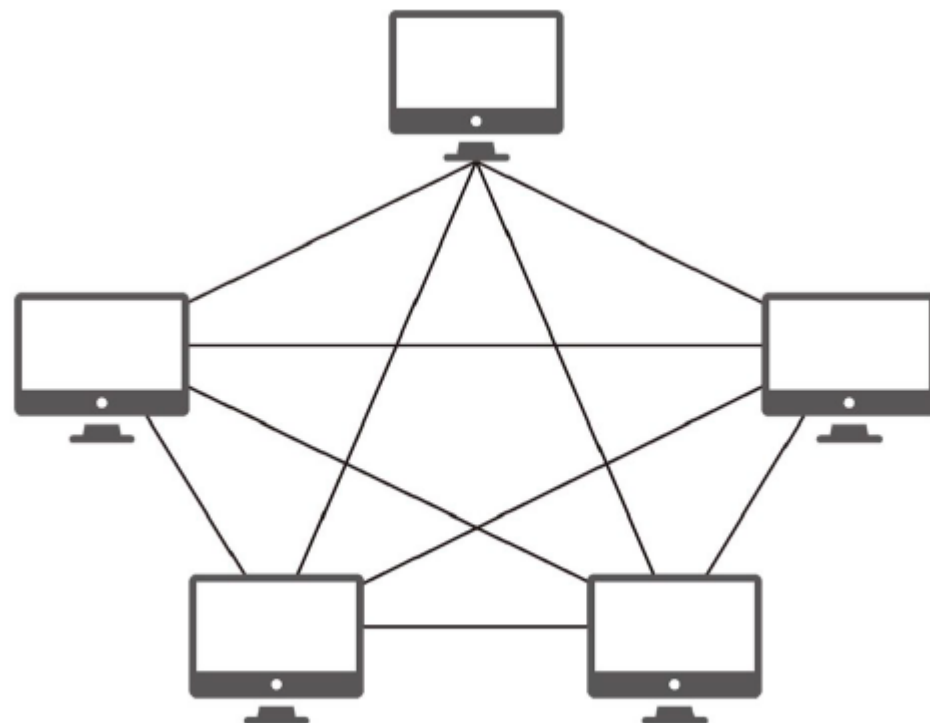
링형 토폴로지



▲ 그림 2-8 링형 토폴로지

- 각각의 노트가 양 옆의 두 노드와 연결하여 전체적으로 고리처럼 하나의 연속된 길을 통해 통신을 하는 망 구성 방식
- 데이터는 노드에서 노드로 이동하며, 각각의 노드는 고리 모양의 길을 통해 패킷 처리
- 장점
 - 노드 수가 증가되어도 네트워크 상의 손실이 거의 없음
 - 충돌이 발생하는 가능성이 적음
 - 노드의 고장 발견이 쉬움
- 단점
 - 네트워크 구성 변경이 어려움
 - 회선에 장애가 발생하면 전체 네트워크에 영향을 크게 끼침

메시 토폴리지



▲ 그림 2-9 메시 토폴로지

- 망형 토폴리지라고도 하며 그물망처럼 연결되어 있는 구조
- 장점
 - 한 단말 장치에 장애가 발생해도 여러 개의 경로가 존재하므로 네트워크를 계속 사용할 수 있다
 - 트래픽 분산 처리 가능
- 단점

- 노드의 추가가 어렵다
- 구축 비용과 운용 비용이 고가이다

병목 현상

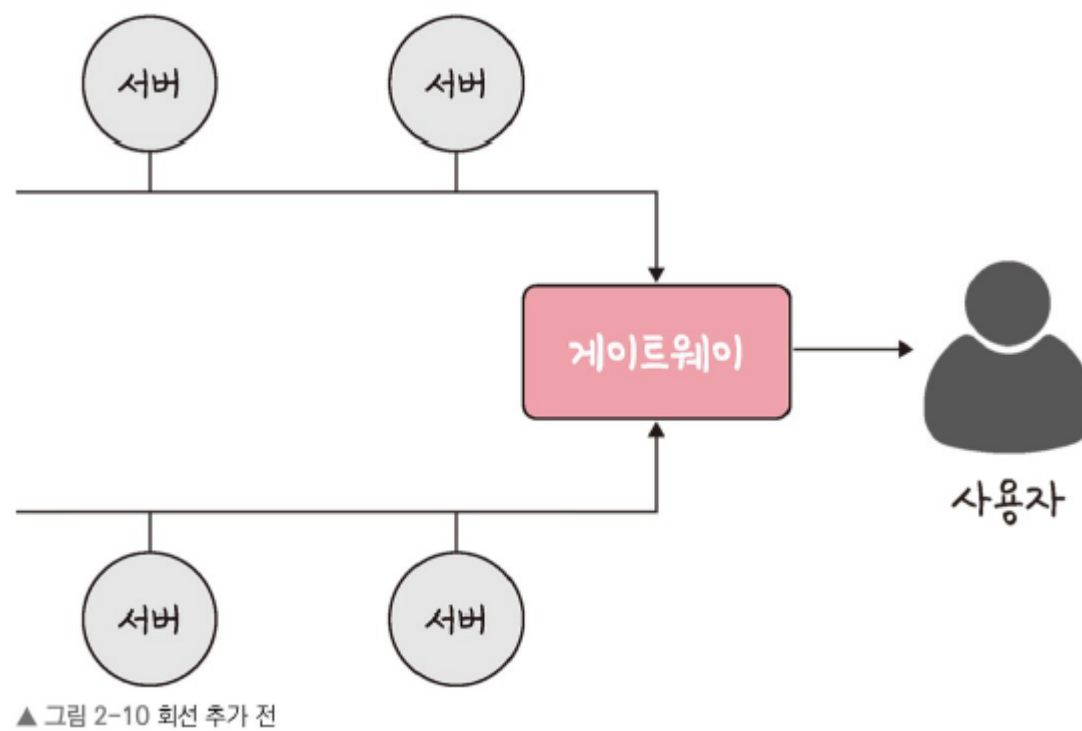
네트워크 토폴로지가 중요한 이유는 병목현상을 찾을 때 중요한 기준이 되기 때문이다.

병목현상이란?

전체 시스템의 성능이나 용량이 하나의 구성 요소로 인해 제한을 받는 현상

병목 현상이 일어났을 경우, 네트워크가 어떤 토폴로지를 갖는지, 또 어떠한 경로로 이루어져 있는지를 안다면 병목 현상에 좀 더 효율적으로 대처할 수 있다.

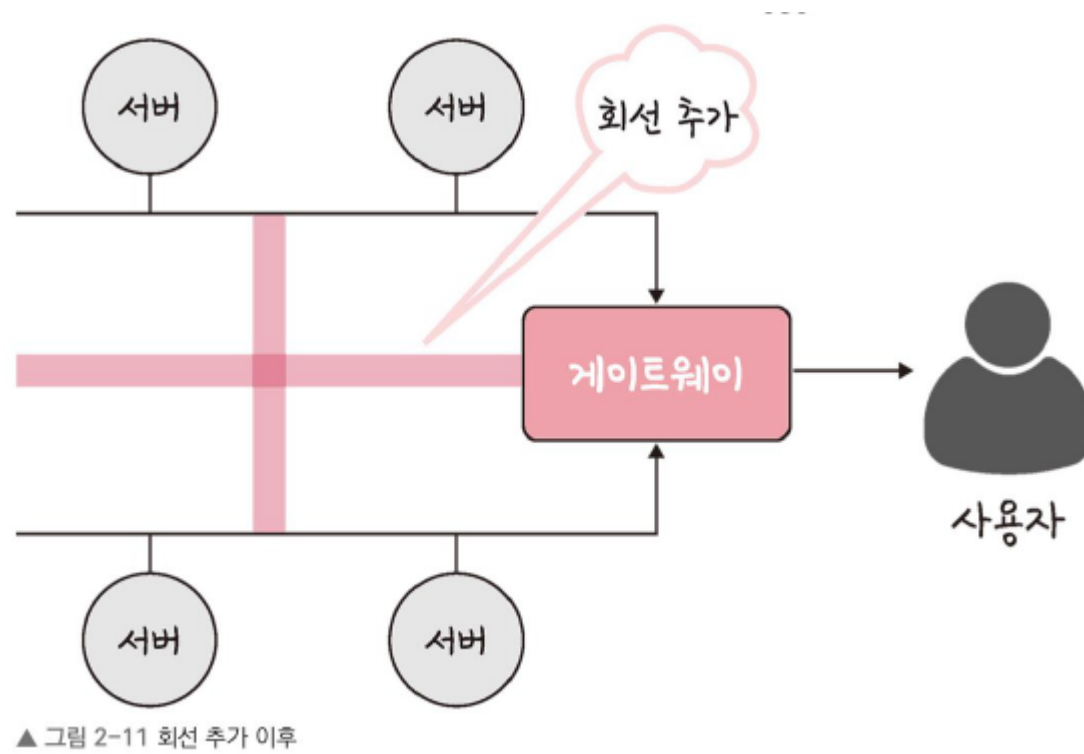
예시



위와 같은 네트워크 토폴로지 상태에서 병목현상이 일어났다고 가정하자.

이런 상황에서 네트워크가 어떤 토폴로지를 갖는지 안다면 회선을 적절히 추가시킴으로서 병목 현상을 해결할 수 있다.

👉 해결법



서버 간 그리고 게이트웨이로 이어지는 회선을 추가하여 병목 현상 해결

참고

<https://blog.naver.com/brickbot/220418950837>

면접을 위한 CS 전공지식 노트