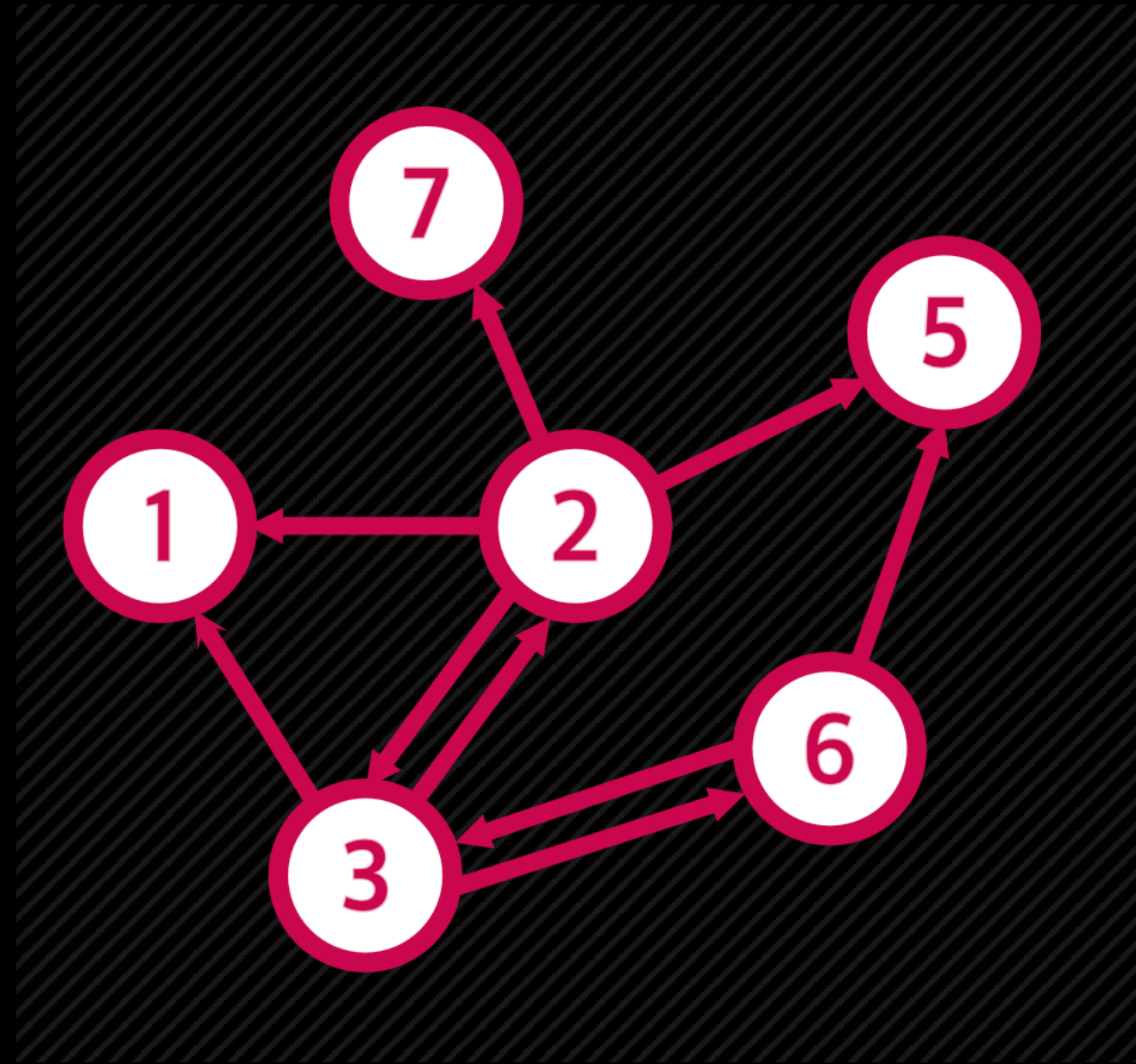


# 그래프 탐색

BFS DFS

# 그래프 탐색

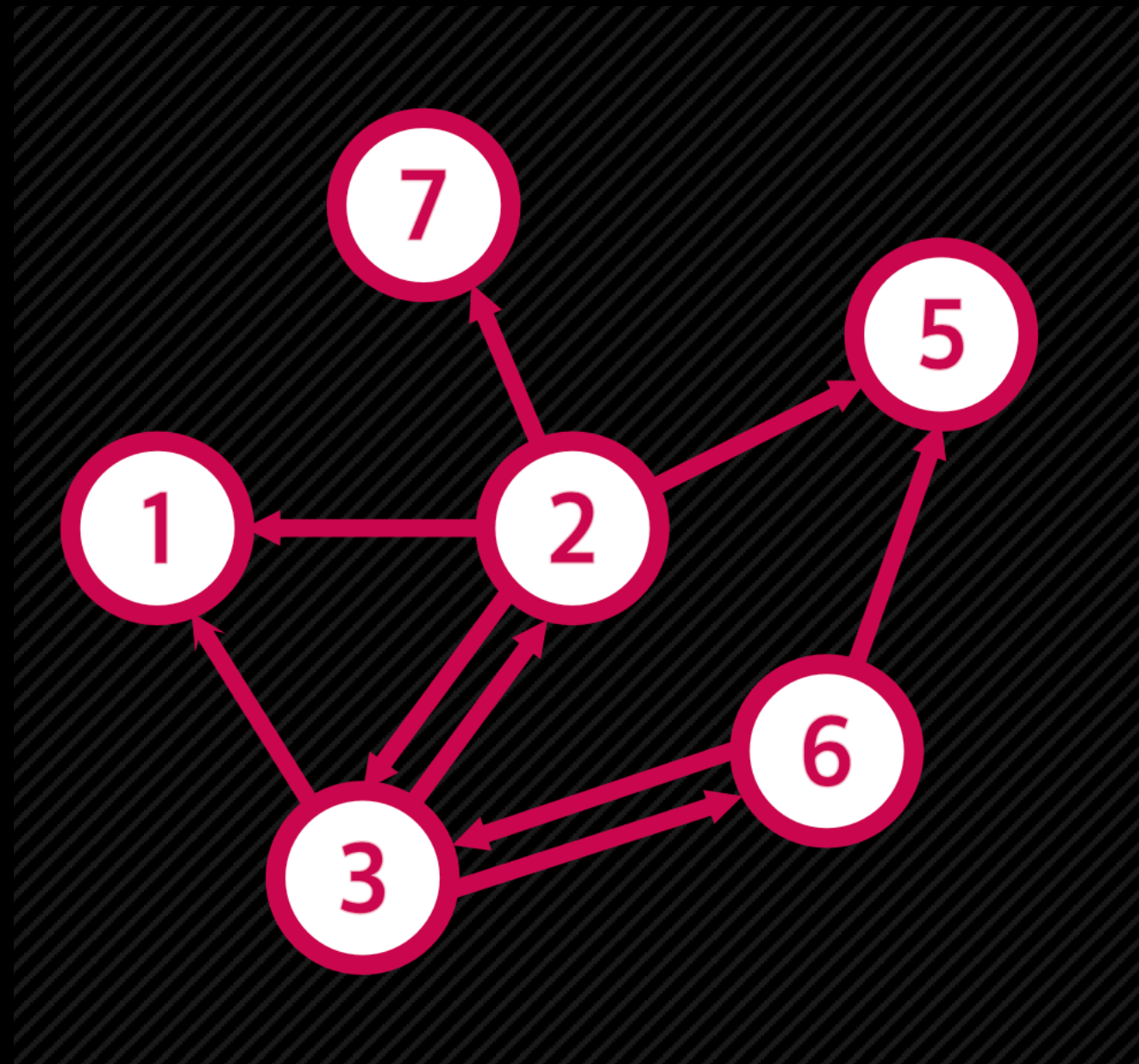
## Graph traversal



그래프의 모든 노드를 탐색하기 위해 간선을 따라 순회하는 것.  
탐색 방법에 따라 BFS / DFS 로 나뉜다.  
그래프의 다양한 응용문제는 탐색을 기반으로 이루어진다.

# 그래프 탐색

## Graph traversal



### Breadth First Search

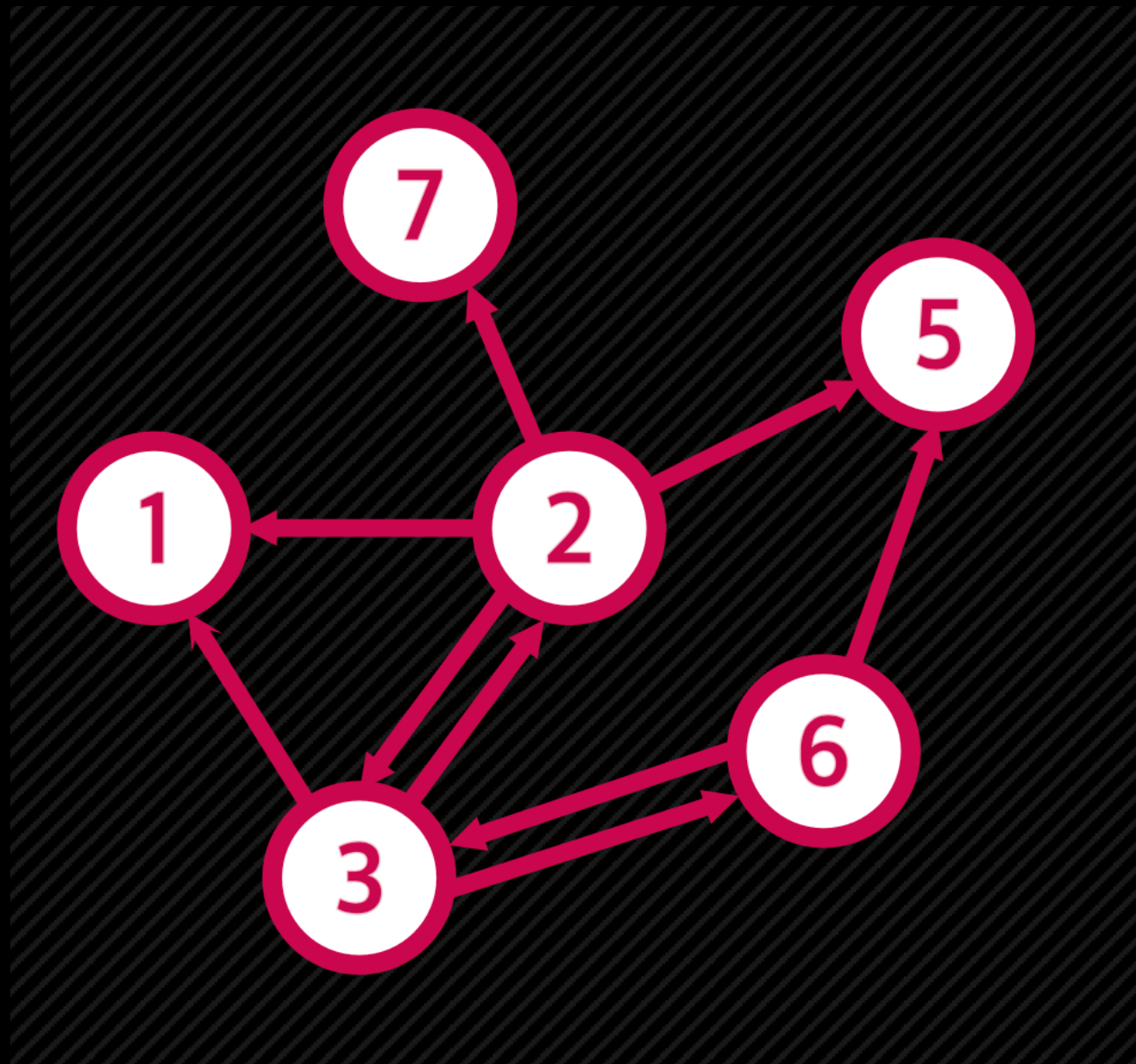
너비 우선 탐색이라고 하며 자신의 자식들부터 순차적으로 탐색  
순차 탐색 이후 다른 자식 노드의 자식을 확인하기 위해 **큐** 사용

### Depth First Search

깊이 우선 탐색이라고 하며, 최대한 깊게 탐색 후 빠져나옴  
백트래킹의 일종이며 **재귀** 활용

# 그래프 탐색

## Graph traversal

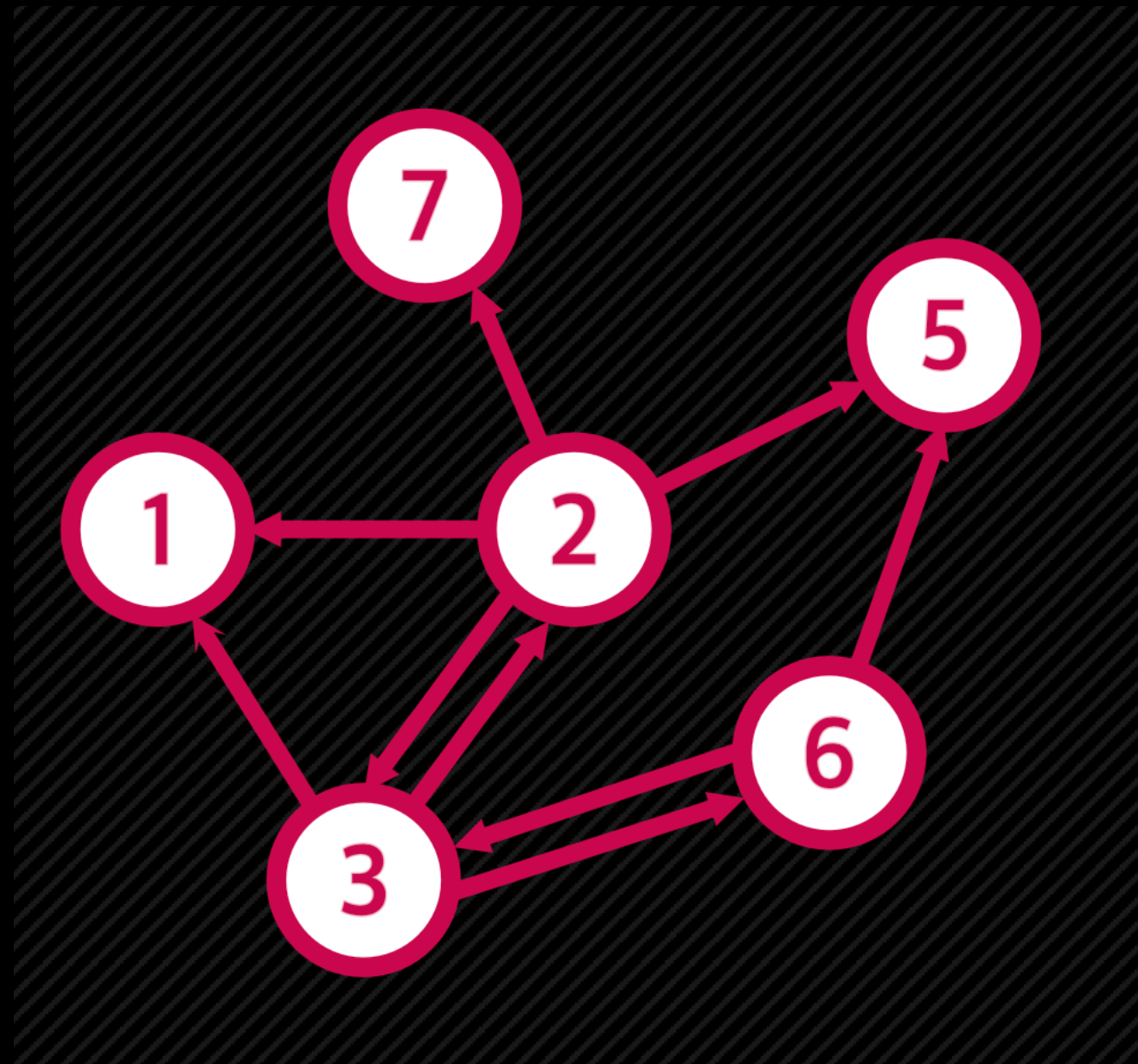


```
function bfs(pos) {  
  set Q = Queue  
  pos -> Q  
  while Q is not empty  
    set node = popped element of Q  
    for children of pos  
      if each child has not been visited  
        visit child  
        child -> Q  
}  
  
function dfs(pos) {  
  visit pos  
  for children of pos  
    if each child has not been visited  
      dfs(pos)  
}
```



# BFS

## Breadth First Search



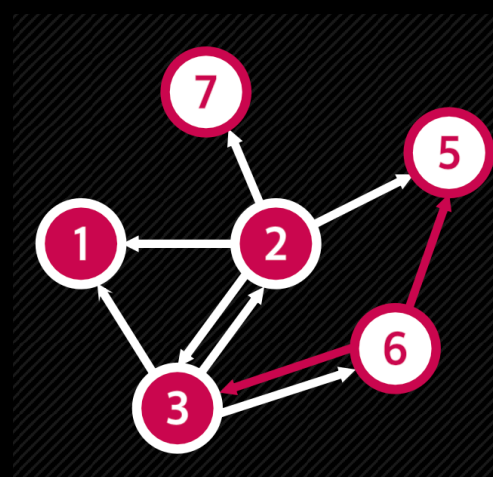
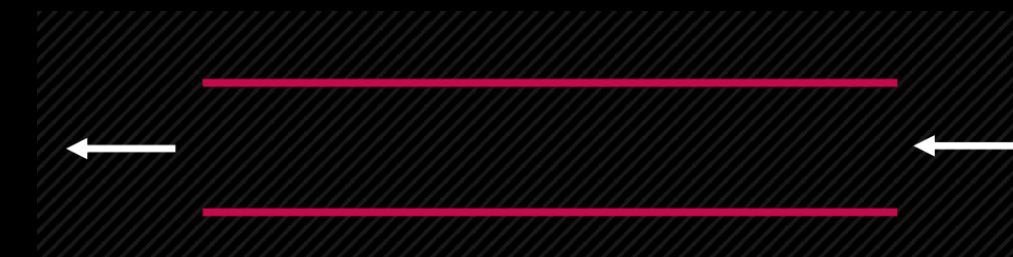
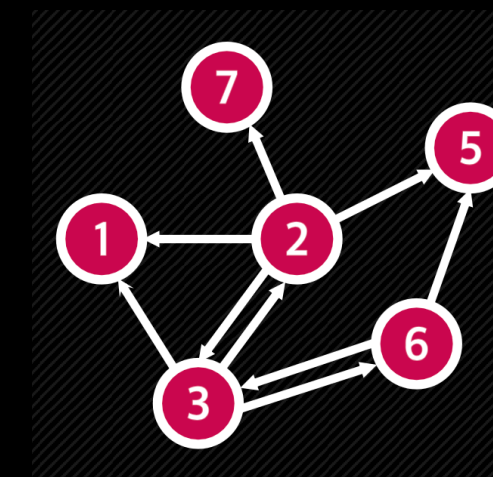
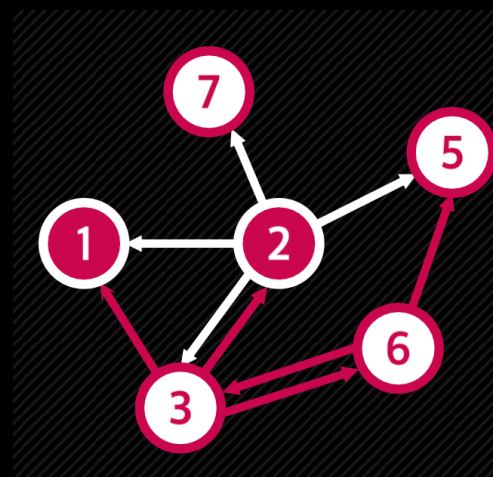
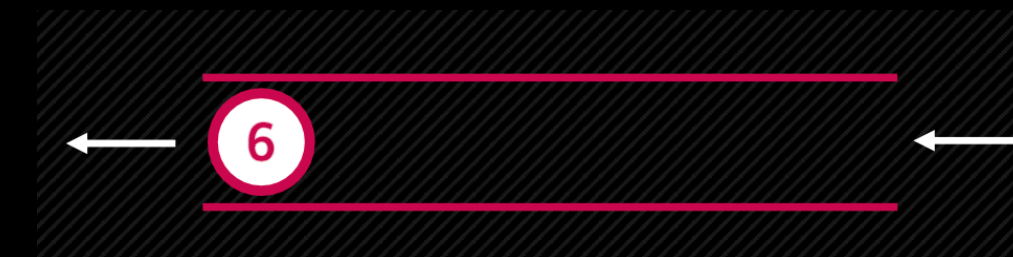
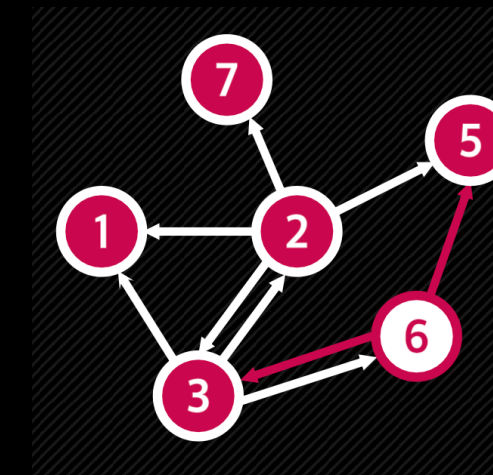
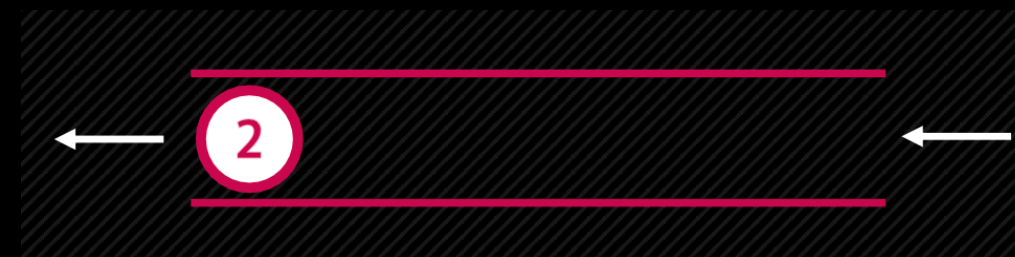
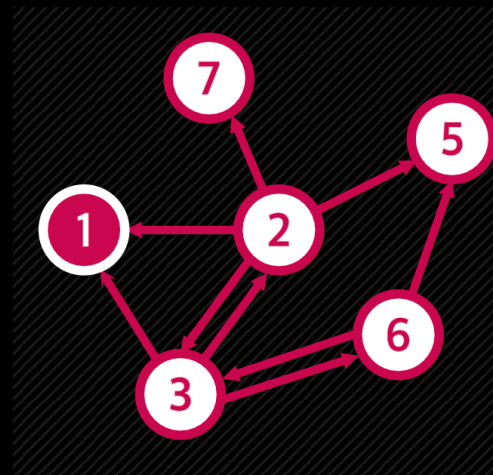
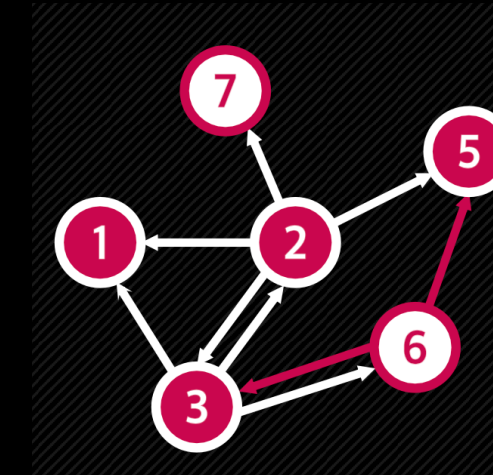
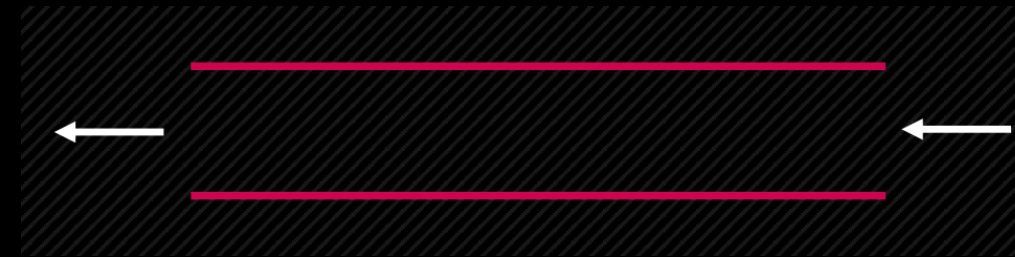
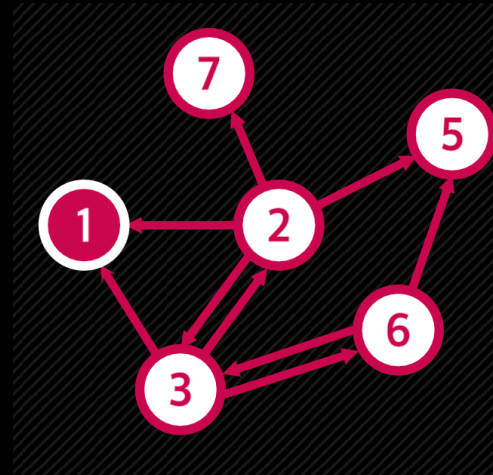
```
function bfs(pos) {  
  set Q = Queue  
  pos -> Q  
  while Q is not empty  
    set node = popped element of Q  
    for children of pos  
      if each child has not been visited  
        visit child  
        child -> Q  
}
```



\* 탐색을 꼭 1번부터 할 필요는 없음. 일반적으로 문제에서 특정 노드부터 탐색할 것을 알려줌

# BFS

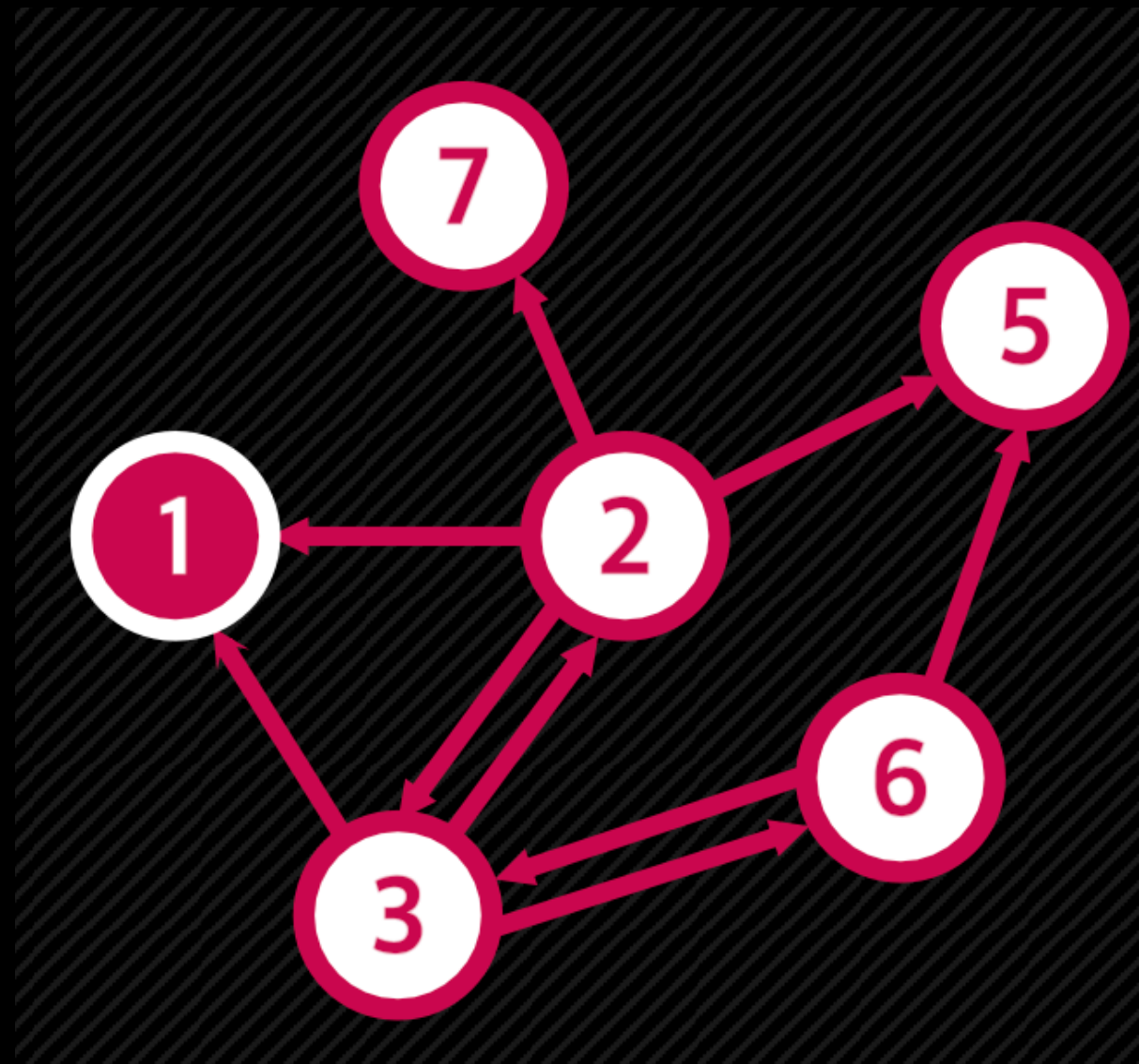
## Breadth First Search





# DFS

## Depth First Search



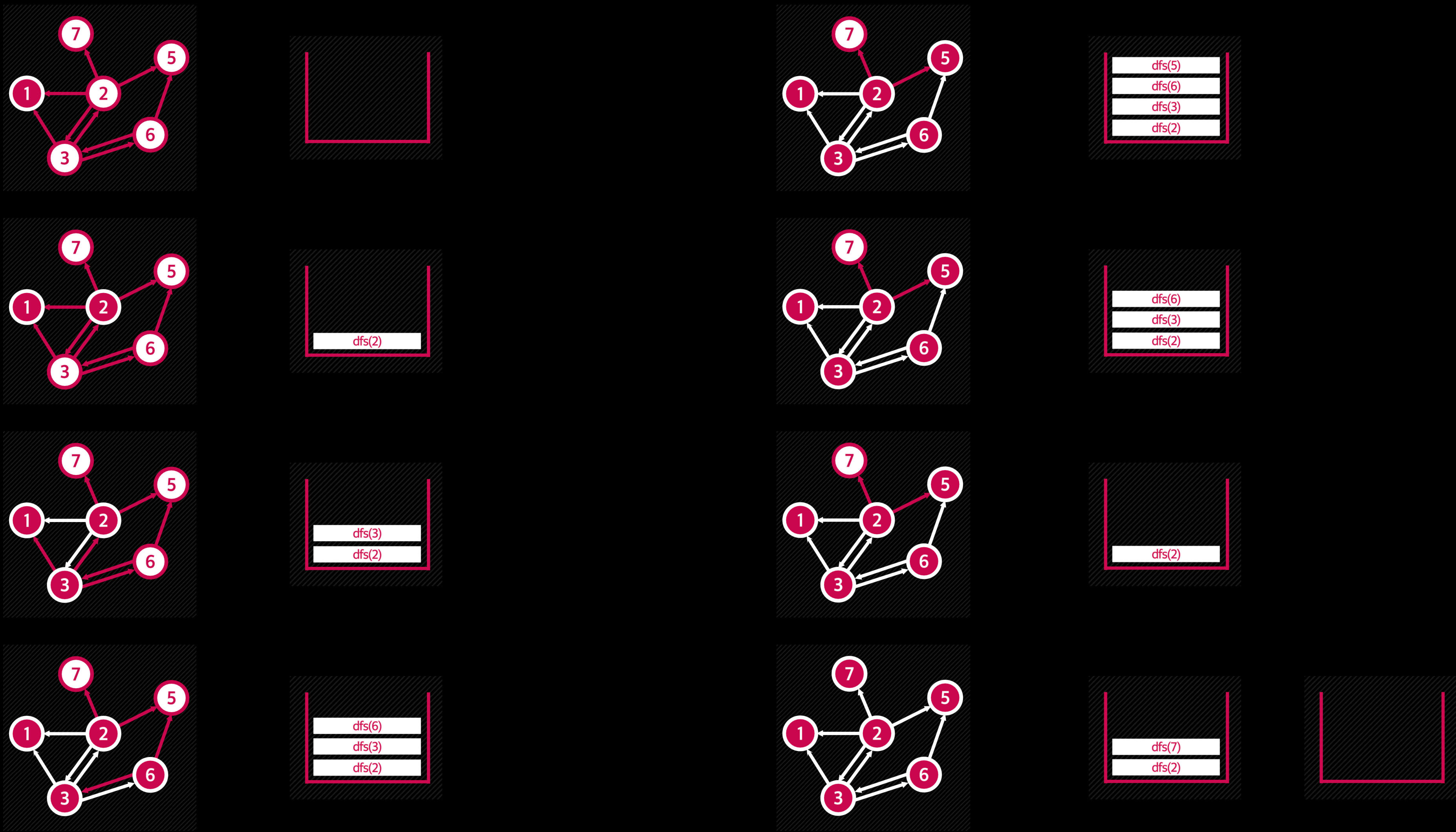
```
function dfs(pos) {  
  visit pos  
  for children of pos  
    if each child has not been visited  
      dfs(pos)  
}
```



\* 탐색을 꼭 1번부터 할 필요는 없음. 일반적으로 문제에서 특정 노드부터 탐색할 것을 알려줌

# DFS

## Depth First Search





# 기초 문제

알고리즘 스터디원들에게는 너무 쉽죠

## 실버 2- BFS와 DFS (#1260)

### 요약

그래프를 BFS/DFS로 탐색한 결과 출력

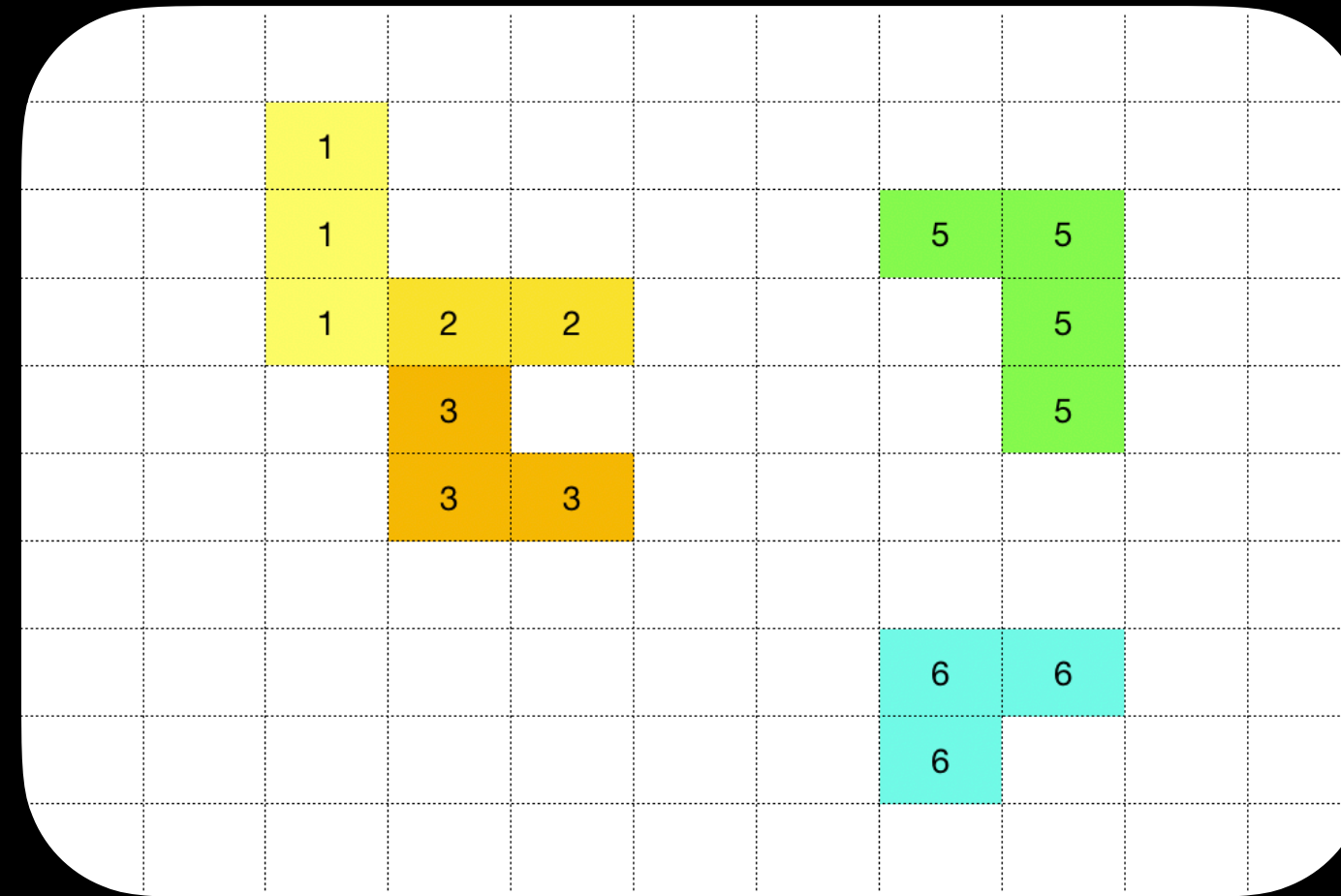
## 실버 3- 바이러스 (#2606)

### 요약

한 컴퓨터가 웜 바이러스에 걸리면 네트워크에 연결된 모든 컴퓨터 같이 걸림  
서로 연결된 정보 주어질 때, **1번을 통해** 감염된 컴퓨터의 수를 구하시오.

# FloodFill

## FloodFill Algorithm



배열에서 어떤 칸과 연결된 영역을 찾는 알고리즘

→ 스택, DFS, BFS로 풀이 가능

 실버 2 - 단지번호 붙이기 (#2667)

 골드 2 - 벽부수고 이동하기 4 (#16946)

 골드 3 - 다리만들기 (#2146)

# 최단 거리

## Shortest Path Problem

 실버 1 - 숨바꼭질 (#1697)

```
if visited[next] == -1:  
    visited[next] = visited[prev] + 1
```

근데 이건 안되지롱

 골드 5 - 숨바꼭질 3 (#13549)

```
if visited[prev] < visited[prev * 2]:  
    visited[prev * 2] = visited[prev]
```



끗

들어주셔서 감사합니다👍