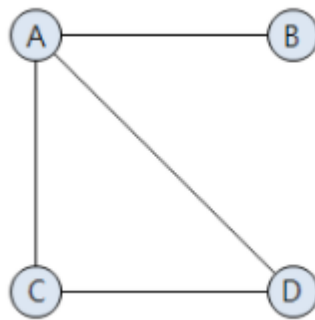




# Week 6

## Graph

- graph란?
  - 정점(Vertex)과 간선(Edge)로 이루어진 자료구조



- 정점(Vertex) : 노드(Node), 데이터가 저장되는 곳
  - 간선(Edge) : 정점(노드) 간의 관계
  - 방향/무방향 그래프, 가중치 그래프, 완전 그래프 등이 있음
- Graph와 Tree의 차이점
  - Tree는 Graph의 한 종류

	Graph	Tree
Cycle 존재 가능 여부	O	X
루트 노드	X	O
부모-자식 관계	X	O

- Graph를 표현하는 방법들
  - 인접 리스트(Adjacency list) : 각 꼭짓점이 인접한 꼭짓점들의 리스트를 가짐
  - 인접 행렬(Adjacency matrix) : Graph의 노드를 2차원 배열로 구현한 것

전체 정점 개수 :  $n$   
전체 간선 개수 :  $e$

	인접 행렬	인접 리스트
간선( $u, v$ ) 검색	$O(1)$	$O(\text{degree}(v))$
정점( $v$ )의 차수 계산	$O(n)$	$O(\text{degree}(v))$
전체 노드 탐색	$O(n^2)$	$O(e)$
메모리	$n \times n$	$n + e$
구현	비교적 easy	비교적 difficult

## DFS

- DFS란?
  - 특정 노드에서 시작해 다음 분기로 넘어가기 전에 해당 분기를 완벽하게 탐색하는 방법
  - Stack 또는 재귀함수로 구현
- DFS를 응용하여 풀 수 있는 문제
  - 경로의 특징을 기억해야 하는 문제
  - 검색 대상의 Graph가 클 때
  - Graph의 주기를 감지

## BFS

- BFS란?
  - 특정 노드에서 시작하여 인접한 노드를 먼저 탐색해나가는 방법
  - Queue로 구현
- BFS를 응용하여 풀 수 있는 문제
  - 최단거리
  - 미로 찾기
  - 촌수 계산
  - 검색 시작 지점으로부터 찾으려는 대상이 멀지 않을 때