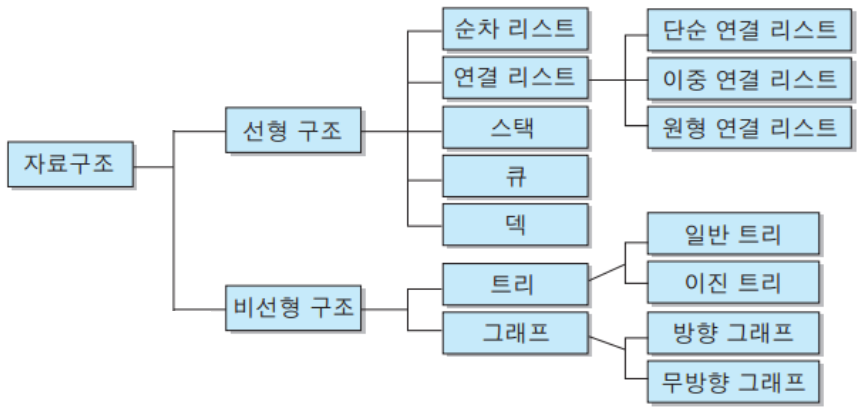
[직무 면접\_자료구조]



먼저 자료구조란 ?

* 컴퓨터 과학에서 효율적인 접근 및 수정을 가능케 하는 자료의 조직, 관리, 저장을 의미한다.
* 적절하게 자료구조를 선택하면 보다 효율적인 알고리즘을 사용할 수 있게 한다.
* 즉 효과적으로 설계된 자료구조는 실행시간과 메모리 용량과 같은 자원을 최소한으로 사용하면서 연산을 수행할 수 있도록 해준다.

자료구조는 크게 선형구조와 비선형 구조로 나뉩니다.

- 선형 구조는 데이터가 연속적으로 연결되어 있는 모양, 즉 원소들을 순차적으로 나열시킨 형태를 의미 합니다.

- 비선형 구조는 하나의 자료 뒤에 여러 개의 자료가 존재할 수 있는 것을 의미합니다. 자료들 간의 앞뒤 관계가 1:n 또는 n:n의 관계를 가집니다.

- **트리**와 **그래프**가 대표적이며 계층적 구조를 나타내기에 적절합니다.

* 트리란 ?

: 노드와 간선으로 구성되어 있고 사이클이 없는 구조를 의미합니다.

* 관련 용어
  + 노드(Node) : 트리를 구성하는 기본 원소
  + 근노드(Root Node) : 부모가 없는 노드
  + 노드의 차수(Degree) : 하위 노드의 개수
  + 트리의 차수 : 트리의 최대 노드 차수
  + 단말 노드(Leaf) : 자식이 없는 노드
  + 노드의 깊이(Level) : 트리의 특정 깊이를 가지는 노드의 집합
* 트리의 종류
  + 이진 트리 : 차수가 2 이하로 구성된 트리
    - 최대 노드의 수는 2의 n-1 승
  + 포화 이진트리 : 모든 레벨에서 노드들이 모두 채워져 있는 트리
  + 완전 이진트리 : 마지막 레벨을 제외하고 모든 노드가 다 채워져 있는 트리(왼쪽 부터 순차적으로 채워지며 마지막 레벨은 다 있을 수도 아닐 수도)
  + 편향 이진트리 : 높이 h에 대한 최소 개수의 노드를 가지면서 한쪽 방향의 자식 노드만을 가진 이진트리
  + 최소 신장트리 (Minimum Spanning Tree) : 그래프 내의 모든 정점을 포함하면서 가중치의 합이 최소인 트리
    - n-1개의 간선을 사용하고 사이클이 포함되면 안된다.
    - 사용 사례 : 통신망, 도로망, 유통망에서 길이, 구축 비용, 전송 시간 등을 최소로 구축하려는 경우 사용
  + MST 구현
    - Kruskal 알고리즘 :
    - Prim 알고리즘
  + 힙 : 완전 이진 트리의 일종으로 우선순위 큐를 위하여 만들어진 자료구조

최소 힙과 최대 힙이 있음

* 이진트리의 특징
  + 높이가 h인 포화 이진 트리(full binary tree)는 2^h-1개의 노드를 가진다
  + 노드가 N개인 포화(full) 혹은 완전(complete) 이진 트리의 *높이*는 O(logN)이다.
  + 노드가 N개인 이진트리의 *높이*는 최악의 경우 O(N)이 될 수 있다.
* 트리의 순회방법
  + 전위순회 : M L R
  + 중위순회 : L M R
  + 후위순회 : L R M
* 트리의 특징
  + N개의 노드에서는 N-1개의 링크를 가진다.
  + 트리에서, 루트에서 어떤 노드로 가는 경로는 유일하다.
* 그래프(Graph)란?  
   : 정점(노드)와 간선으로 구성되어 있으며 사이클이 존재하는 비선형 자료 구조입니다.
* 관련 용어
  + 정점(vertex) : 위치 라는 개념(node 라고도 부름)
  + 간선(edge) : 위치 간의 관계. 즉, 노드를 연결하는 선 (link, branch)
  + 인접 정점(adjacent vertex) : 간선에 의해 직접 연결된 정점
  + 정점의 차수(degree) : 무방향 그래프에서 하나의 정점에 인접한 정점의 수

(무방향 그래프에 존재하는 정점의 모든 차수의 합 = 그래프의 간선 수의 2배)

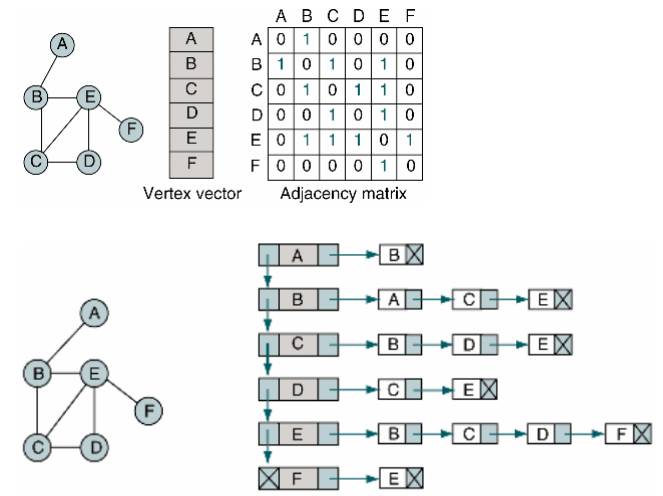
* + 진입 차수(in-degree) : 방향 그래프에서 외부에서 오는 간선의 수
  + 진출 차수(out-degree) : 방향 그래프에서 외부로 향하는 간선의 수

(방향 그래프에 있는 정점의 진입 차수 또는 진출 차수의 합

= 방향 그래프의 간선의 수)

* + 경로 길이 : 경로를 구성하는 데 사용된 간선의 수
  + 단순 경로 : 경로 중에서 반복되는 정점이 없는 경우
  + 사이클 : 단순 경로의 시작 정점과 종료 정점이 동일한 경우
* 그래프의 특징
  + 그래프는 네트워크 모델 이다.
  + 2개 이상의 경로가 가능하다.
  + 즉, 노드들 사이에 무방향/방향에서 양방향 경로를 가질 수 있다.
  + self-loop 뿐 아니라 loop/circuit 모두 가능하다.
  + 루트 노드라는 개념이 없다.
  + 부모-자식 관계라는 개념이 없다.
  + 순회는 DFS나 BFS로 이루어진다.
  + 그래프는 순환(Cyclic) 혹은 비순환(Acyclic)이다.
  + 그래프는 크게 방향 그래프와 무방향 그래프가 있다.
  + 간선의 유무는 그래프에 따라 다르다.

* 그래프의 종류
  + 방향/무방향 그래프
  + 연결/비연결 그래프
  + 가중치 그래프 : 네트워크라고도 한다. ex)도로의 연결, 도로의 길이
  + 순환/비순환 그래프
  + 완전그래프 : 그래프의 모든 정점이 연결되어 있는 그래프  
    (정점의 수가 n일 때 간선의 수는 n\*(n-1)/2 개)
* 그래프의 구현
  + 인접 리스트(Adjacency List) : 각각의 정점에 인접한 정점들을 연결 리스트로  
    (간선의 수가 적을 때 사용) 모든 간선의 수 O(N+E)
  + 인접 행렬(Adjacency Matrix) : N\*N의 행렬을 만들고 연결되면 1 or true로 표현  
    (간선의 수가 많을 때 사용) 모든 간선의 수 O(N^2)



* 그래프의 탐색
  + 깊이 우선 탐색(DFS, Depth-First Search) : 재귀
  + 너비 우선 탐색(BFS, Breadth-First Search) : 큐
* 트리와 그래프의 차이

