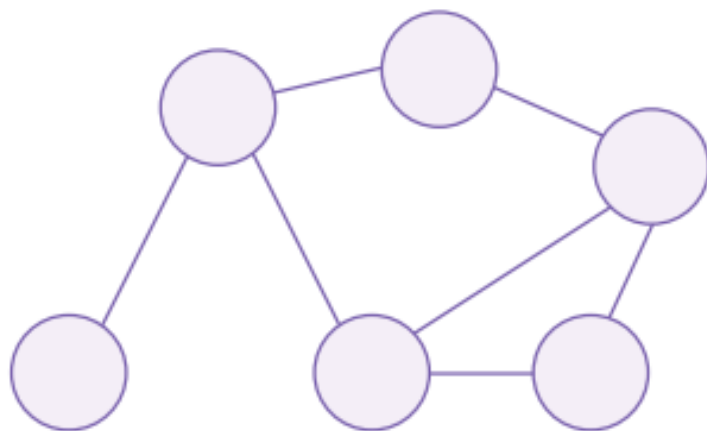


Python Algorithm

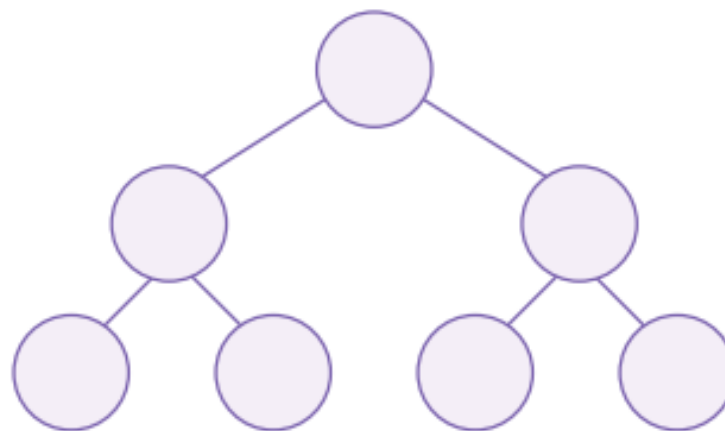
비선형 자료구조

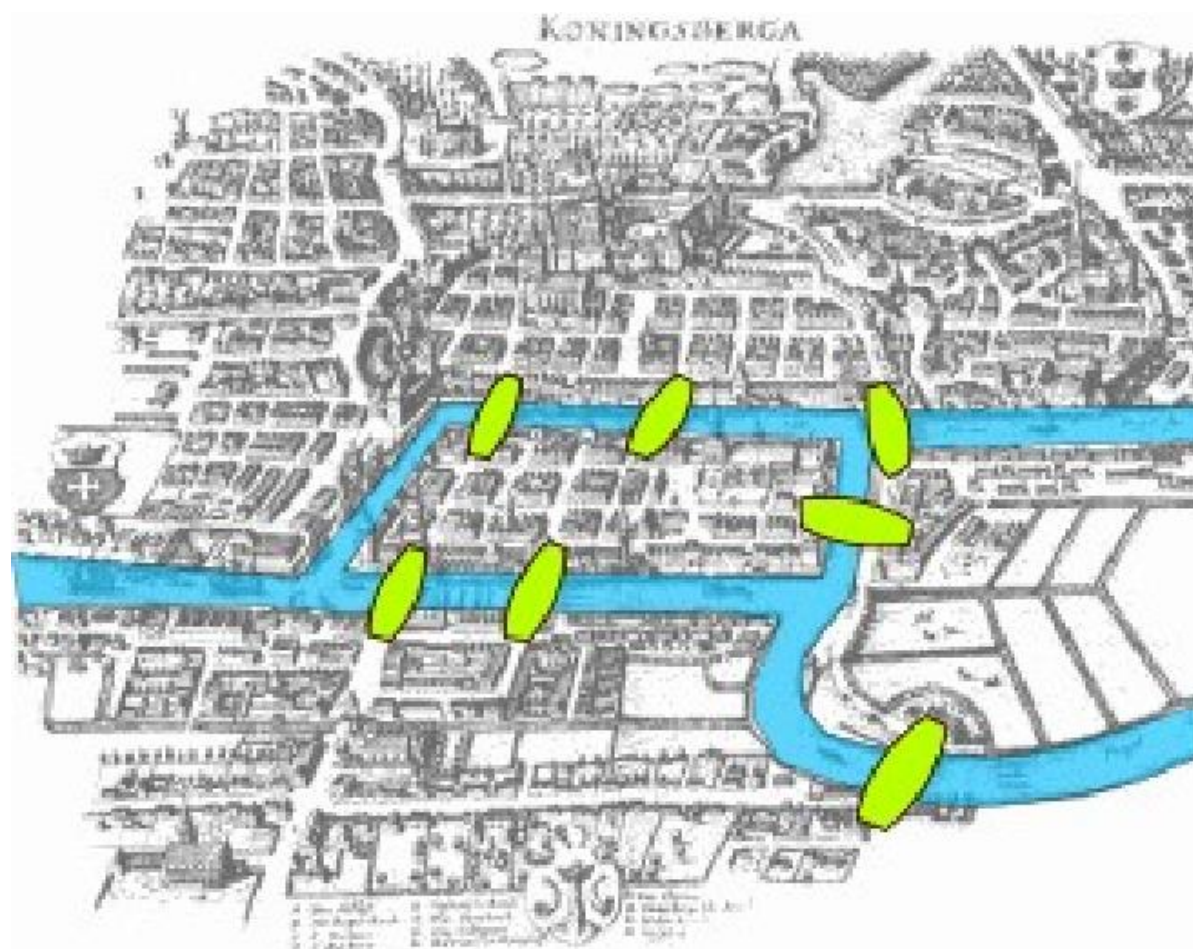
데이터 요소가 순차적으로 배열되지 않는 자료구조

그래프



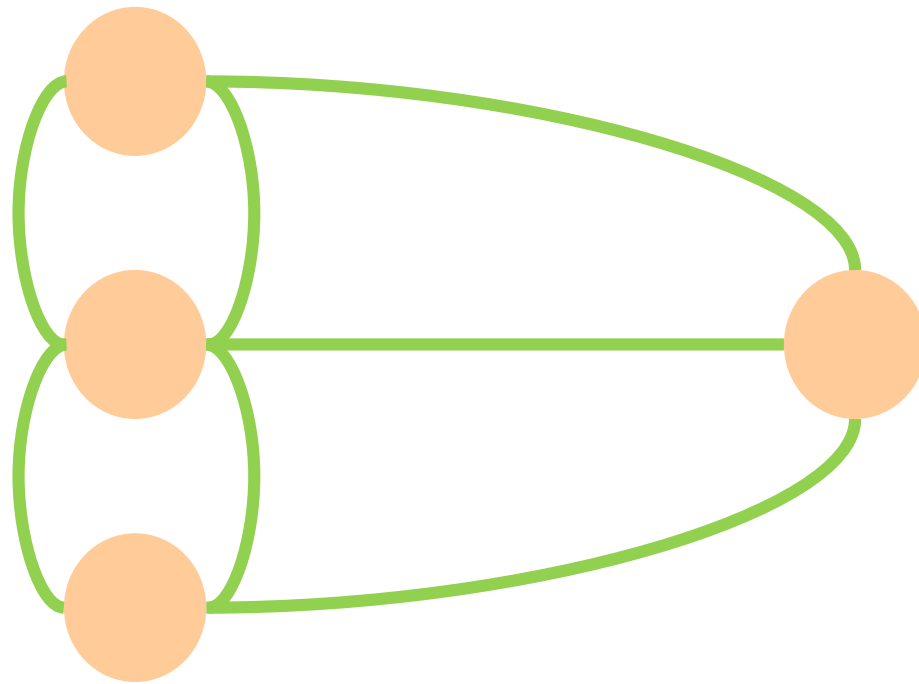
트리





비선형 자료구조 - 그래프

노드와 각 노드를 연결하는 간선으로 구성된 자료 구조

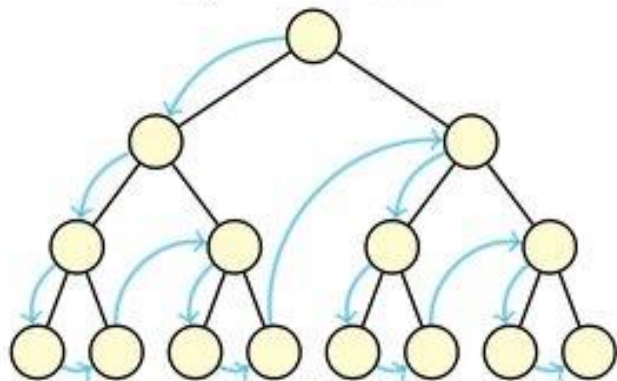


비선형 자료구조 - 그래프의 특징

항목	설명
연결 방법	어떤 방향이든 가능 (양방향, 단방향, 무방향)
구성	어떻게 연결되어도 상관없음 (순환 구조 가능)
시작 노드의 기준	어느 쪽에서 시작해도 무방
다른 노드와의 관계	수평 관계
순회 방법	DFS, BFS

비선형 자료구조 - 그래프 순회(그래프 탐색)

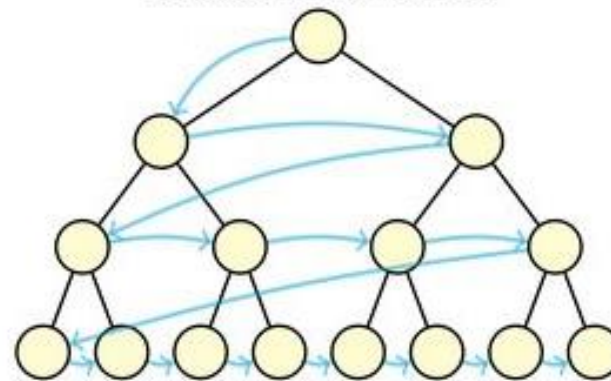
Depth-First Search



깊이 우선 탐색 (DFS)

- 스택 또는 재귀를 통해 구현
- 정답이 몇 가지인지 물어보는 문제에 적합

Breadth-First Search

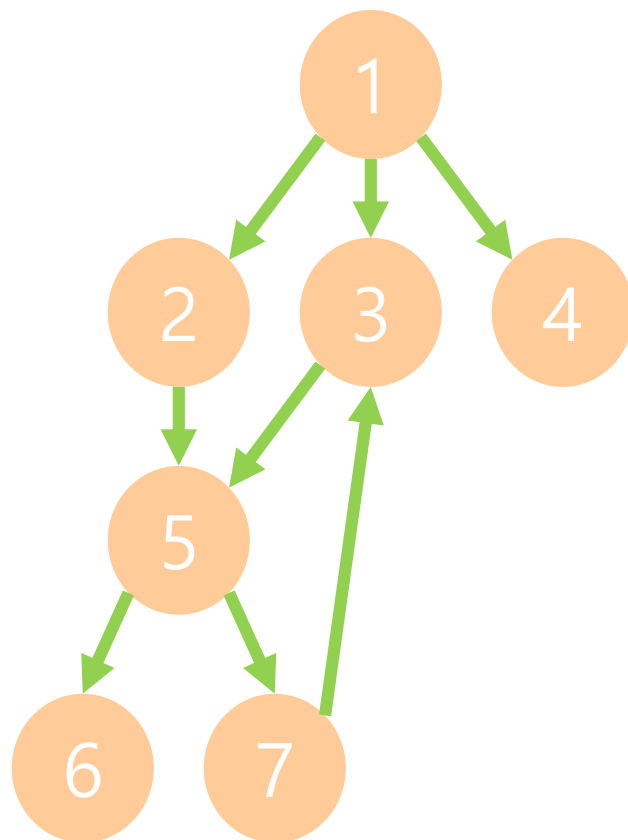


너비 우선 탐색 (BFS)

- 큐를 이용해 구현
- 최단 경로를 찾는 문제에 적합

비선형 자료구조 - 그래프의 표현 (인접 행렬)

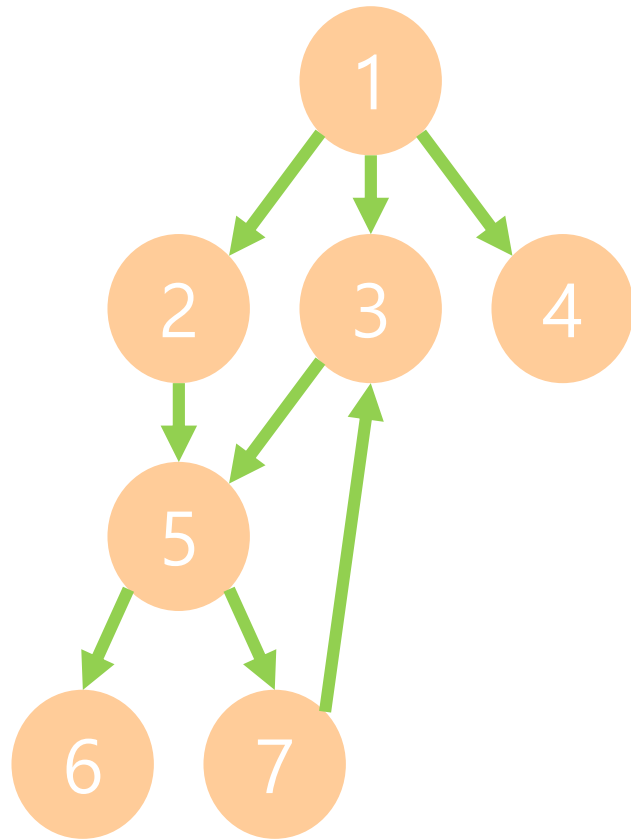
인접 행렬은 2차원 노드의 개수만큼 2차원 배열을 만듭니다.



	1	2	3	4	5	6	7
1	0	1	1	1	0	0	0
2	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	1	1
6	0	0	0	0	0	0	0
7	0	0	1	0	0	0	0

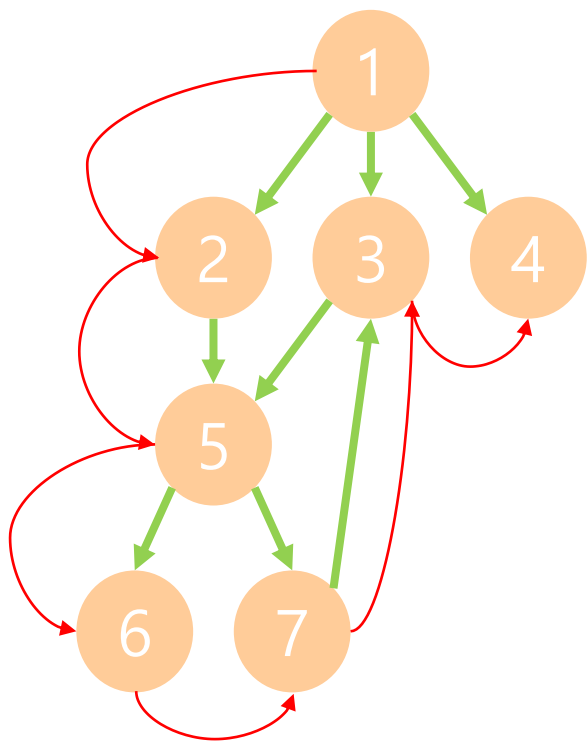
비선형 자료구조 - 그래프 표현 (인접 리스트)

인접 리스트는 출발 노드를 키로, 도착 노드를 값으로 표현합니다.



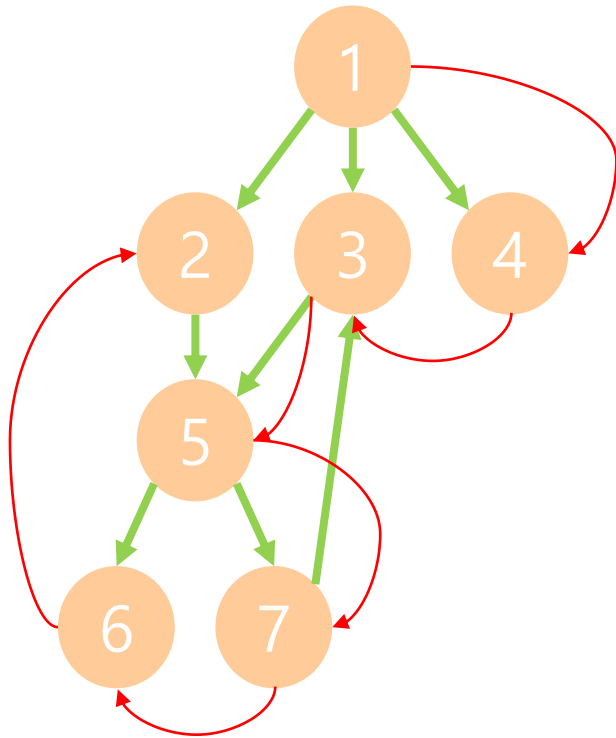
```
graph = {  
    1 : [2, 3, 4],  
    2 : [5],  
    3 : [5],  
    4 : [],  
    5 : [6, 7],  
    6 : [],  
    7 : [3]  
}
```


비선형 자료구조 - 재귀구조로 구현한 깊이 우선 탐색 (DFS)



```
def recursive_dfs(v, discovered=[]):  
    discovered.append(v)  
    for w in graph[v]:  
        if not w in discovered:  
            discovered = recursive_dfs(w, discovered)  
    return discovered  
  
print(recursive_dfs(1))
```

비선형 자료구조 - 스택을 이용해 구현한 깊이 우선 탐색 (DFS)



```
def stack_dfs(start_v):  
    discovered = []  
    stack = [start_v]  
    while stack:  
        v = stack.pop()  
        if v not in discovered:  
            discovered.append(v)  
            for w in graph[v]:  
                stack.append(w)  
    return discovered  
  
print(stack_dfs(1))
```

문제. 부분수열의 합

<https://www.acmicpc.net/problem/1182>

N개의 정수로 이루어진 수열이 있을 때,

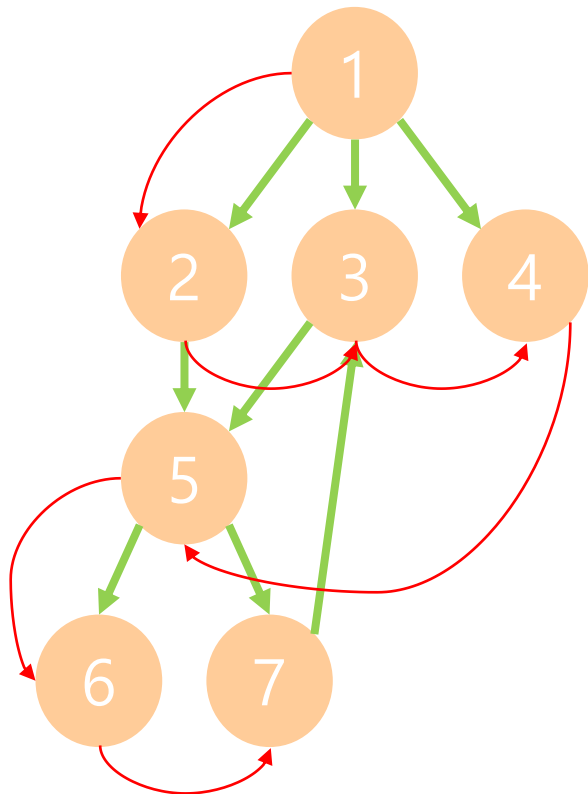
크기가 양수인 부분수열 중에서
그 수열의 원소를 다 더한 값이 S가 되는 경우의 수를 구하는 프로그램을 작성하시오.

입력	출력
5 0 -7 -3 -2 5 8	1

풀이. 부분수열의 합

<https://www.acmicpc.net/problem/1182>

비선형 자료구조 - 너비 우선 탐색 (BFS)



```
from collections import deque

def queue_bfs(start_v):
    discovered = [start_v]
    queue = deque([start_v])
    while queue:
        v = queue.popleft()
        for w in graph[v]:
            if w not in discovered:
                discovered.append(w)
                queue.append(w)
    return discovered

print(queue_bfs(1))
```

문제. 미로 탐색

<https://www.acmicpc.net/problem/2178>

$N \times M$ 크기의 배열로 표현되는 미로가 있다.

미로에서 1은 이동할 수 있는 칸을 나타내고, 0은 이동할 수 없는 칸을 나타낸다.

(1, 1)에서 출발하여 (N, M)의 위치로 이동할 때 지나야 하는 최소의 칸 수를 구하는 프로그램을 작성하시오. 칸을 셀 때에는 시작 위치와 도착 위치도 포함한다.

입력	출력
46 101111 101010 101011 111011	15

풀이. 미로 탐색

<https://www.acmicpc.net/problem/2178>

비선형 자료구조 - 순열

<https://leetcode.com/problems/permutations>

서로 다른 정수 배열을 입력받아 가능한 모든 순열을 반환하는 프로그램을 작성하시오.

입력

nums = [1, 2, 3]

출력

```
[  
  [1, 2, 3],  
  [1, 3, 2],  
  [2, 1, 3],  
  [2, 3, 1],  
  [3, 1, 2],  
  [3, 2, 1]  
]
```


비선형 자료구조 - 조합

<https://leetcode.com/problems/combinations>

입력값으로 두 개의 정수 n 과 k 가 주어지면,
1~ n 범위에서 k 개의 조합을 모두 반환하는 프로그램을 작성하시오

입력	출력
$n=4$ $k=2$	<pre>[[1, 2], [1, 3], [1, 4], [2, 3], [2, 4], [3, 4]]</pre>



비선형 자료구조 - 부분 집합

<https://leetcode.com/problems/subsets>

주어진 배열로 구성할 수 있는 가능한 모든 부분 집합을 반환하는 프로그램을 작성하시오.

입력

nums = [1, 2, 3]

출력

```
[  
  [],  
  [1],  
  [1,2],  
  [1,2,3],  
  [1,3],  
  [2],  
  [2,3],  
  [3]  
]
```

비선형 자료구조 - 조합을 통해 숫자 합 구하기

<https://leetcode.com/problems/combination-sum>

숫자 집합 candidates를 조합하여,
합이 target이 되는 원소를 나열하는 프로그램을 작성하시오.

각 원소는 중복으로 나열 가능하다.

입력	출력
<pre>candidates = [2, 3, 6, 7] target = 7</pre>	<pre>[[7], [2, 2, 3]]</pre>