그래프

그래프? 링크에 의해 연결되어 있는 노드들로 구성된 구조

노드(=정점, vertex) , 링크(= 간선, edge)

G=(V,E) : V와 E가 집합이고, E의 모든 원소가 V의 두-원소 부분집합일 때 V의 원소는 정점 , E의 원소는 간선

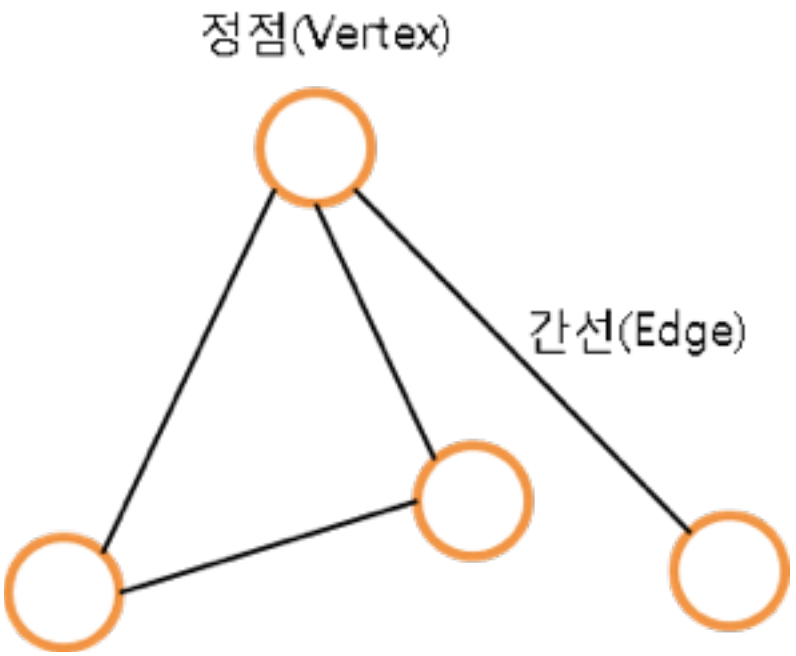
그래프의 크기(size) : 정점의 개수

경로의 길이(length) : 간선의 개수

단순경로 : 모든 정점이 서로 다른 것 { SE, UK, FR, DE, CZ }

도달가능 : 경로가 존재하는 것. 정점 v1은 정점 v0으로부터 도달가능하다.

연결 그래프 : 어떤 정점이 다른 모든 정점으로부터 도달 가능한 것

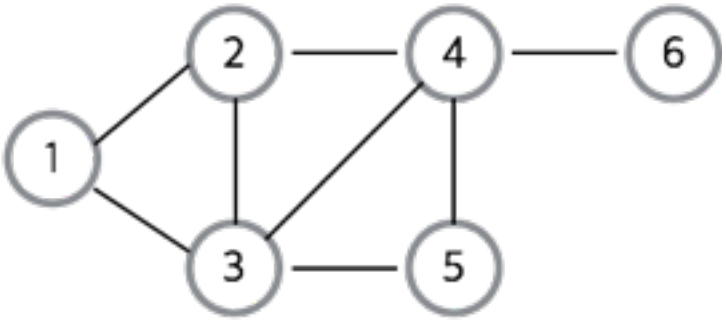


[그래프 저장 방법 – 인접 행렬, 인접 리스트]

(1) 인접 행렬

boolean 원소의 2차원 배열. 하나의 정점 당 하나의 행과 열을 가지고 있음

Ad	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	1	1	0	0
3	1	1	0	1	1	0
4	0	1	1	0	1	1
5	0	0	1	1	0	0
6	0	0	0	1	0	0

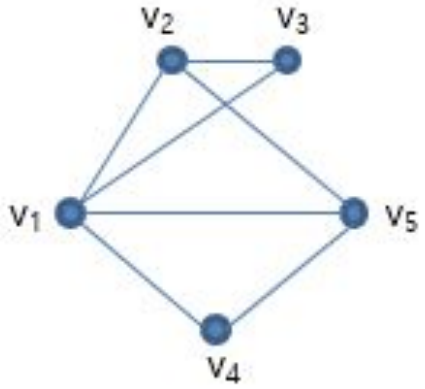


- 참인 항목의 개수는 그래프에 있는 간선 개수의 두배
- 주대각선에 대해 대칭 ‘ a[i][j] = a[j][i] = true ‘
- 1,0으로 표현하면 후에 가중치 그래프로도 표현 가능(기본적으로는 boolean 사용)

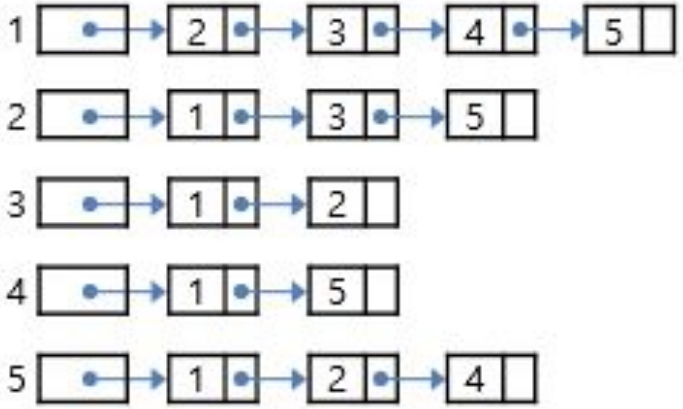
(2) 인접 리스트

각 정점 당 하나의 리스트를 가진 연결 리스트의 배열 ‘ List[] a ‘

‘정점 v를 위한 리스트는 v에 인접한 각 정점에 대한 하나의 노드 포함’



정점	인접 정점
1	2,3,4,5
2	1,3,5
3	1,2
4	1,5
5	1,2,4

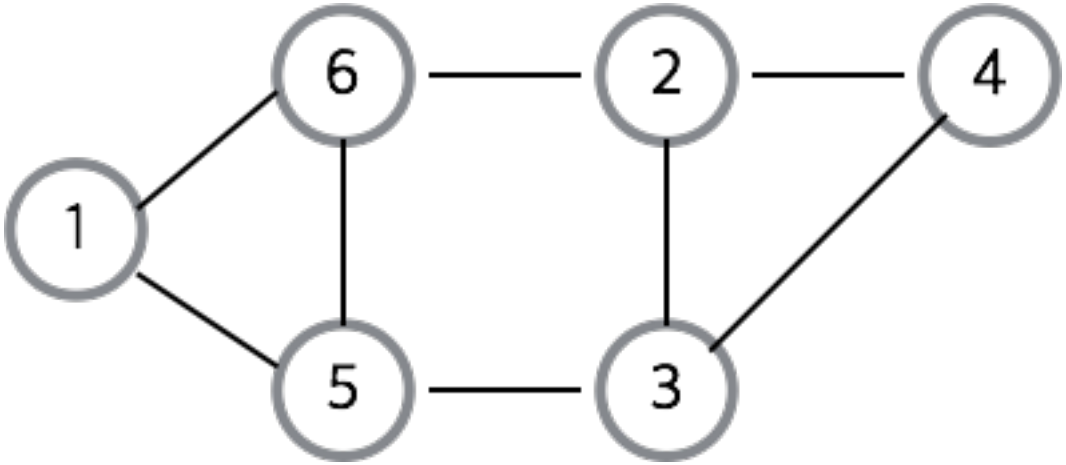


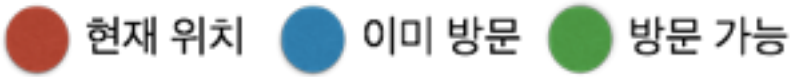
- 인접리스트에 있는 노드의 총 개수는 간선 개수의 두배

```
a [ index(v) ].add( index(w) );
a [ index(w) ].add( index(v) );
```

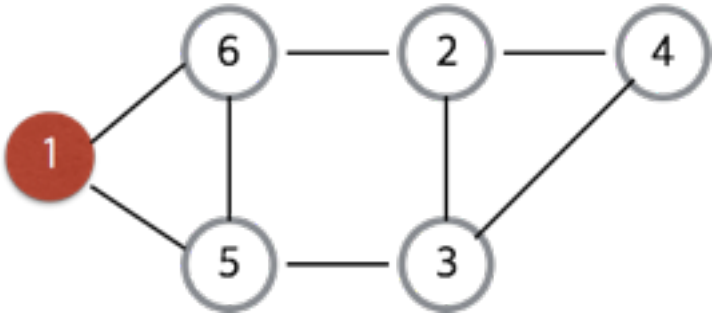
[너비 우선 탐색(BFS)] - ‘Queue’ 사용

그래프를 순회하는 것은 트리를 순회하는 것과 비슷

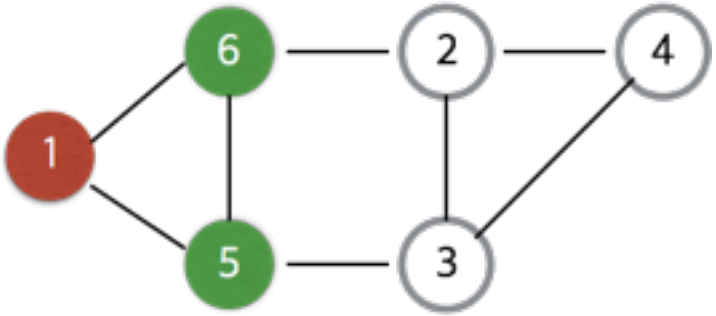




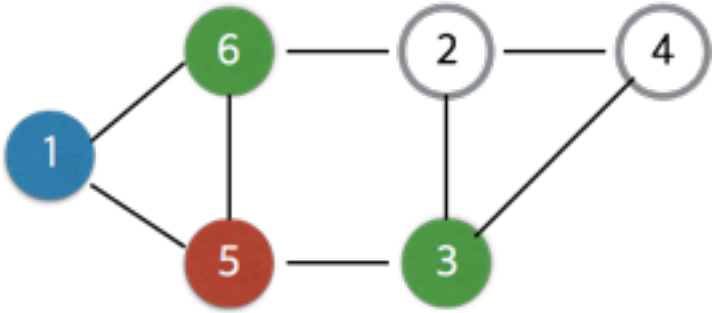
큐	1					
결과						



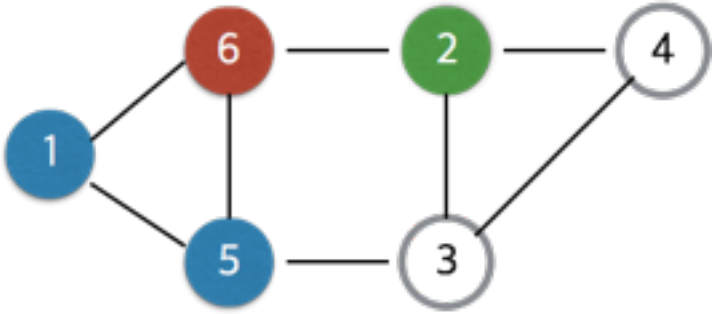
큐		5	6			
결과	1					



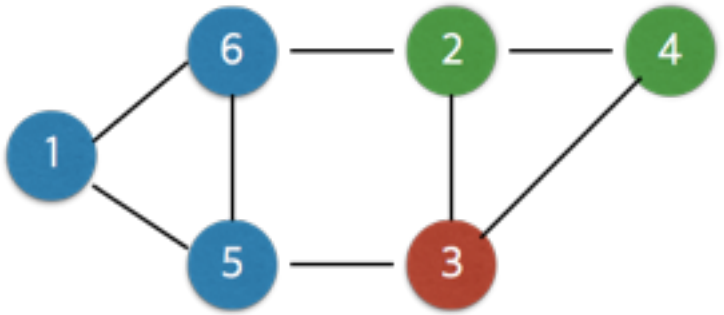
큐			6	3		
결과	1	5				



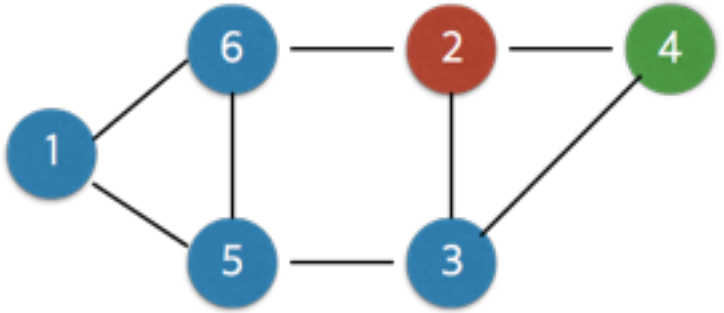
큐				3	2	
결과	1	5	6			



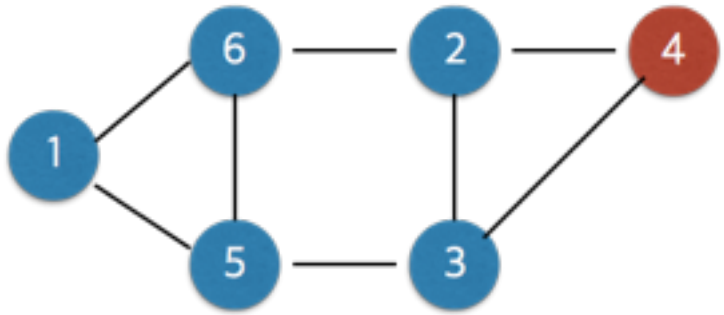
큐					2	4
결과	1	5	6	3		



큐						4
결과	1	5	6	3	2	



큐						
결과	1	5	6	3	2	4

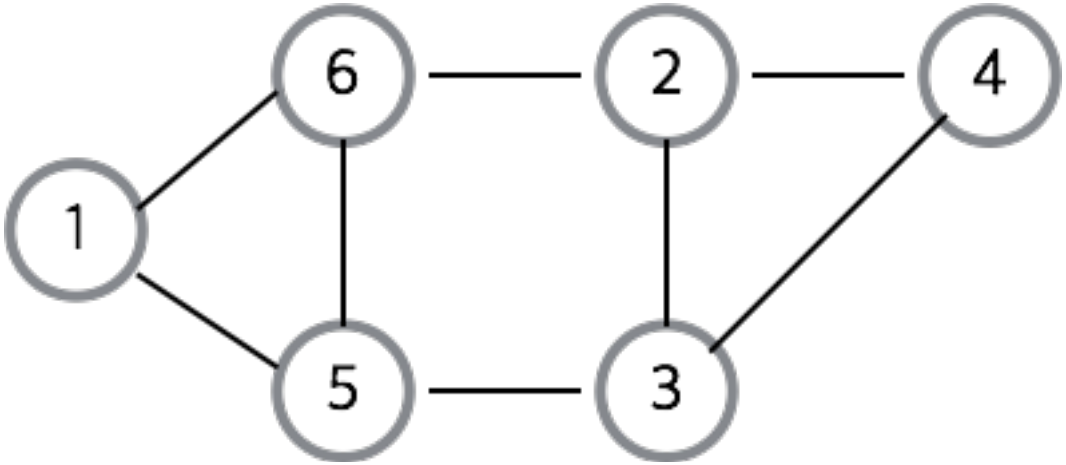


http://manducku.tistory.com/24

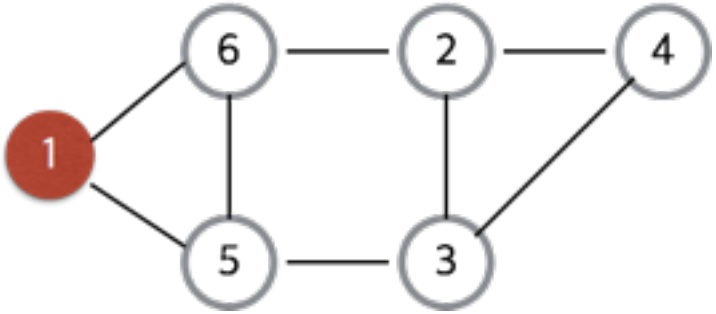
[깊이 우선 탐색(DFS)] - '재귀' 사용

" 갈 수 있을 때까지 간다 " 현재 정점과 인접한 간선들을 하나씩 검사하다가, 아직 방문하지 않은 정점으로 향하는 간선이 있다면 그 간선을 향해 간다

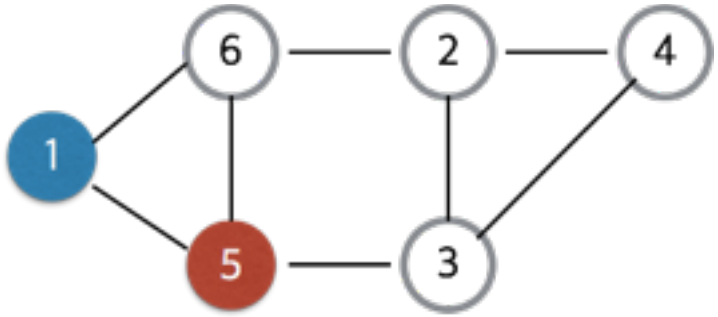
더 이상 갈 곳이 없는 막힌 정점에 도달하면 포기 , 마지막에 따라왔던 간선을 다시 돌아가며 탐색



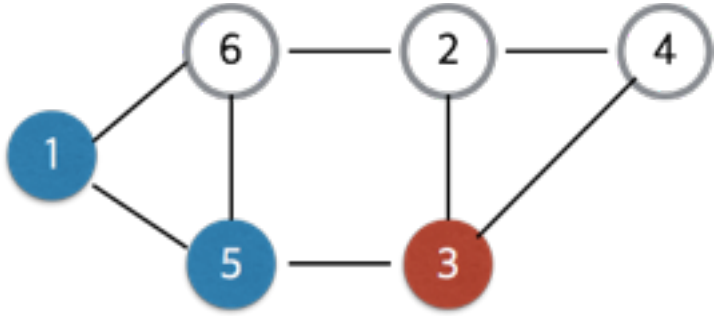
순서	1					
스택	1					



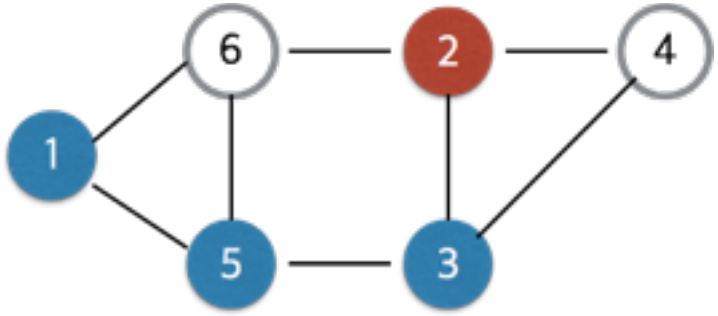
순서	1	5				
스택	1	5				



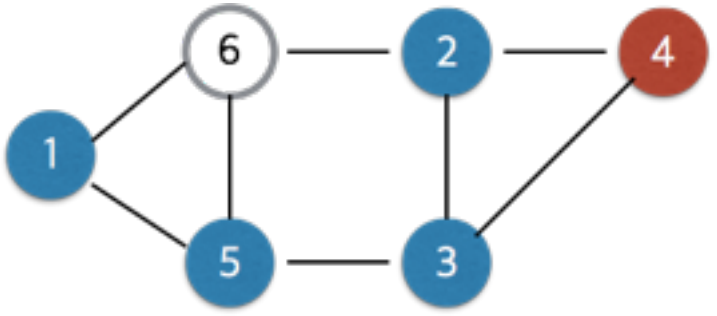
순서	1	5	3			
스택	1	5	3			



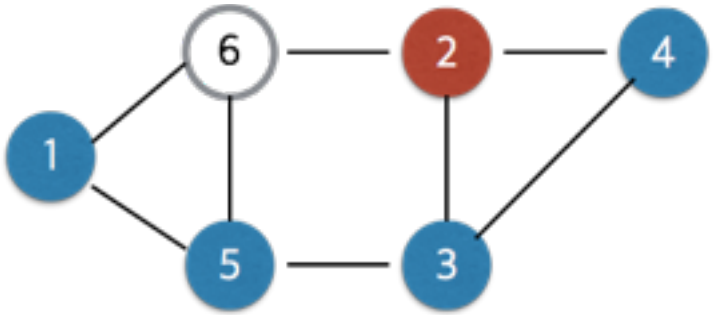
순서	1	5	3	2		
스택	1	5	3	2		



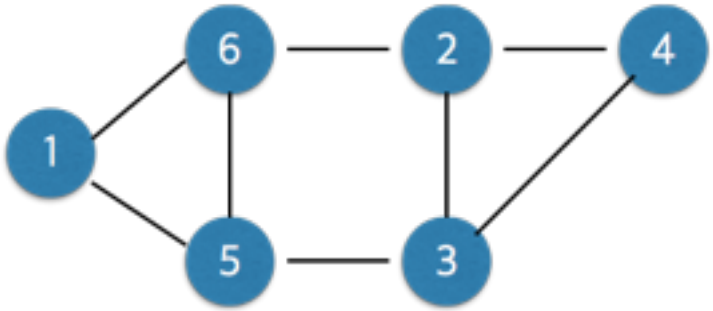
순서	1	5	3	2	4	
스택	1	5	3	2	4	



순서	1	5	3	2	4	
스택	1	5	3	2		



순서	1	5	3	2	4	6
스택	1	5	3	2	6	



깊이 우선 탐색의 중요한 특성 : **더 따라갈 간선이 없을 경우** 이전**으로 돌아가야 한**다

이 것을 구현하기 위에선 지금까지 거쳐온 정점들을 모두 저장해 둬야 하는데, 재귀 호출을 이용하면 이와 같은 일을 간단히 할 수 있음

예시에서는 Stack을 예를 들었으나, 배열을 추가로 2개 만들어야하는 등 고려할 사항이 많기 때문에.. 비 추천

http://manducku.tistory.com/23