

graph

Graph의 정의

그래프 G 는 두 집합 V, E 로 나타낸다

1. $V(G)$: 노드(node) 혹은 정점(vertex)의 집합
2. $E(G)$: 정점을 잇는 에지(edge)의 집합

무방향 그래프(undirected graph)

무방향 그래프(undirected graph)

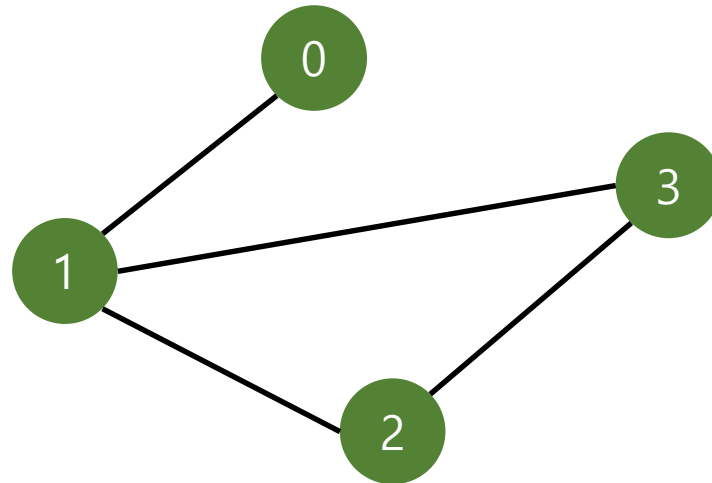
$G=(V, E)$

$V(G)=\{0, 1, 2, 3\}$

$E(G)=\{(0, 1), (1, 2), (2, 3), (1, 3)\}$

정점의 수 : n

최대 에지 수 : $n(n-1)/2$



방향 그래프(directed graph)

방향 그래프(directed graph)

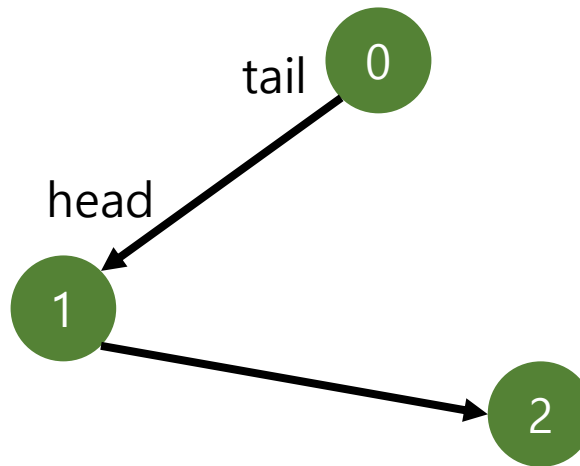
$G = \langle V, E \rangle$

$V(G) = \{0, 1, 2\}$

$E(G) = \{ \langle 0, 1 \rangle, \langle 1, 2 \rangle \}$

정점의 수 : n

최대 에지 수 : $n(n-1)$



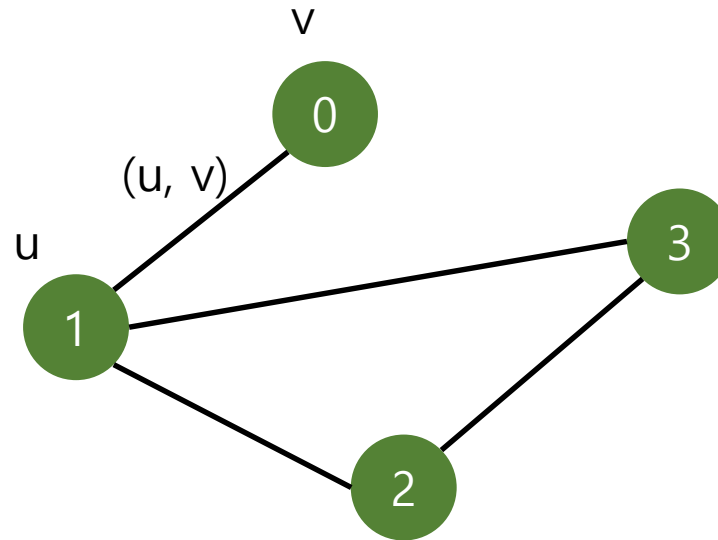
Graph의 특징

1. 자기 간선(self-edge)를 가질 수 없다.
 (v, v) or $\langle v, v \rangle$ 를 가질 수 없다.
2. 두 정점 사이에 같은 에지를 여러 개 가질 수 없다.

다중 그래프(multigraph)

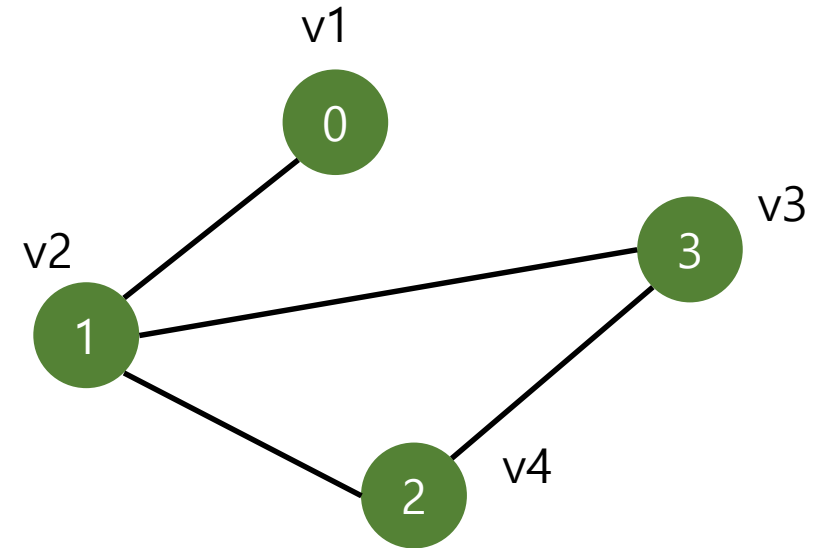
Adjacent와 incident

$(u, v) \in E(G)$ 이면 정점 u 와 정점 v 는 인접한다(adjacent)
에지 (u, v) 는 u 와 v 에 부속된다(incident)



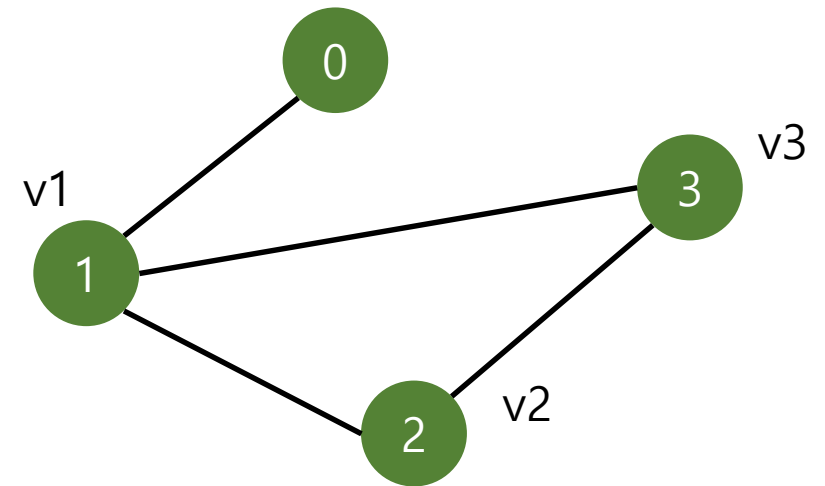
경로(path)와 길이(length)

1. 경로(path)
: $(v1, v2), (v2, v3), (v3, v4)$ 가 집합 $E(G)$ 의 원소일 때
 $v1$ 에서 $v4$ 까지의 정점 순서($v1 \rightarrow v2 \rightarrow v3 \rightarrow v4$)
2. 경로 길이(length)
: 어떤 경로에서 에지의 수
3. **단순 경로(simple path)**
: 어떤 경로에서 처음과 마지막을 제외하고
모든 정점이 다를 때

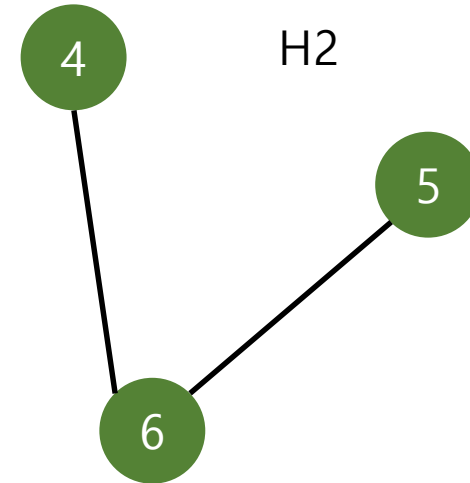
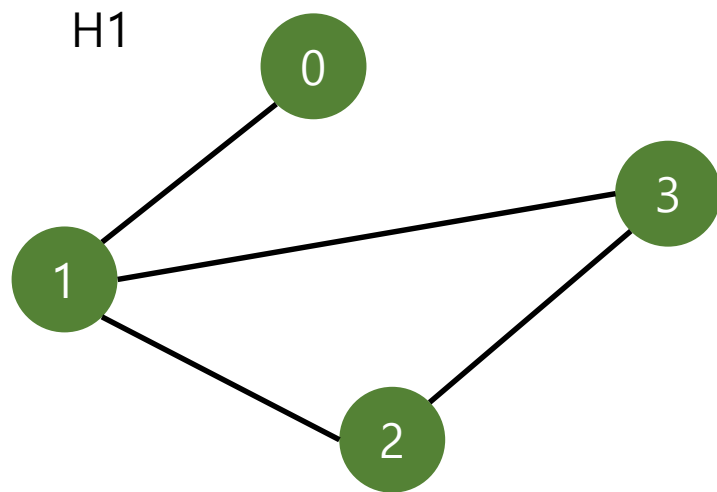


사이클(cycle)과 connected

1. 사이클(cycle)
: 단순 경로에서 처음과 마지막 정점이 같을 때
 $v1 \rightarrow v2 \rightarrow v3 \rightarrow v1$
2. Connected
: 정점 u 에서 정점 v 까지의 경로가 있다면
두 정점은 연결되었다(connected)
3. 연결된 그래프(Connected graph)
: 어떠한 임의의 노드 u 와 v 에 대해서도
연결되어 있다면 그래프 G 가 연결되었다



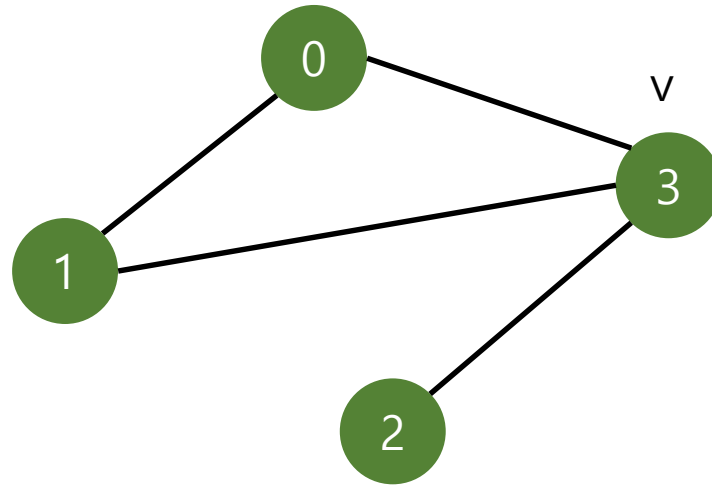
연결 요소(connected component)



차수(degree)

차수(degree)
: 정점 v 에 소속된 에지의 수

$$d(v) = 3$$



In-degree와 out-degree

방향 그래프에서 어떤 정점 v 에 들어오는 에지의 수를
In-degree(진입 차수)라고 한다.

$$\mathbf{in-d(v) = 1}$$

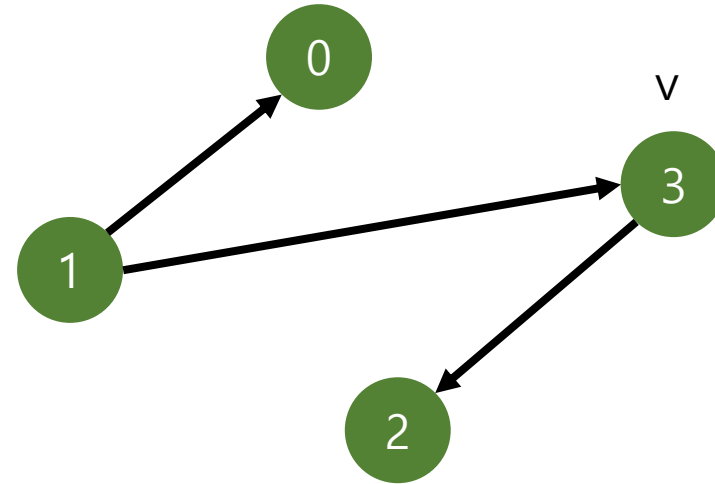
나가는 에지의 수는

Out-degree(진출 차수)라고 한다.

$$\mathbf{out-d(v) = 1}$$

차수(degree)는 $in-deg + out-deg$

$$\mathbf{d(v) = 2}$$

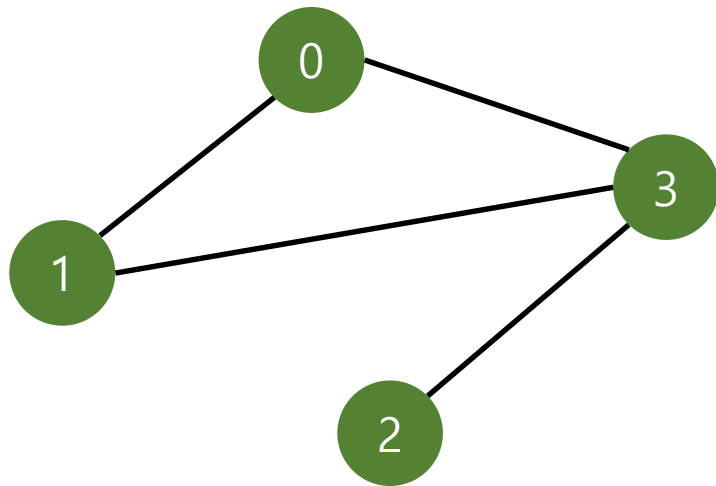


부분 그래프(subgraph)와 부분 신장 그래프(spanning subgraph)

1. 부분 그래프(subgraph)
: $V(G') \subseteq V(G)$ 이고 $E(G') \subseteq E(G)$ 인 그래프 G' 를
그래프 G 의 부분 그래프(subgraph)라고 한다.
2. 부분 신장 그래프(spanning subgraph)
: $V'=V$ 이고 $E(G') \subseteq E(G)$ 인 그래프 G' 를
그래프 G 의 부분 신장 그래프(spanning subgraph)라고 한다.

그래프 표현(graph representation)
: 인접 행렬(Adjacency matrix)

$n \times n$ 행렬 **if $(i, j) \in E(G)$ then 1**
 else 0



Operations

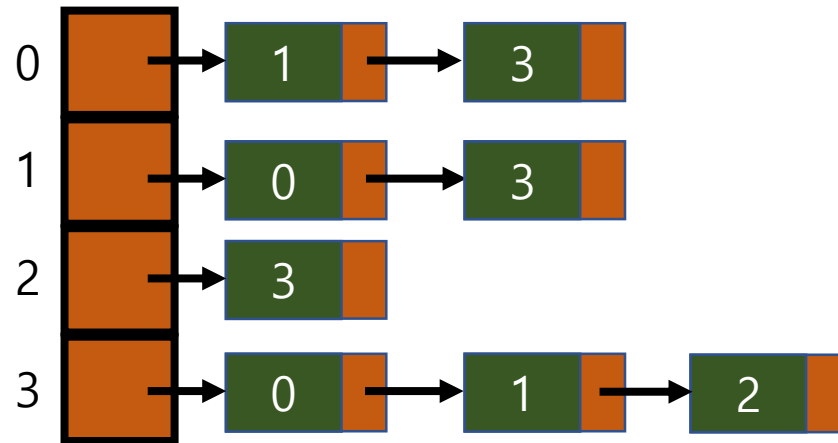
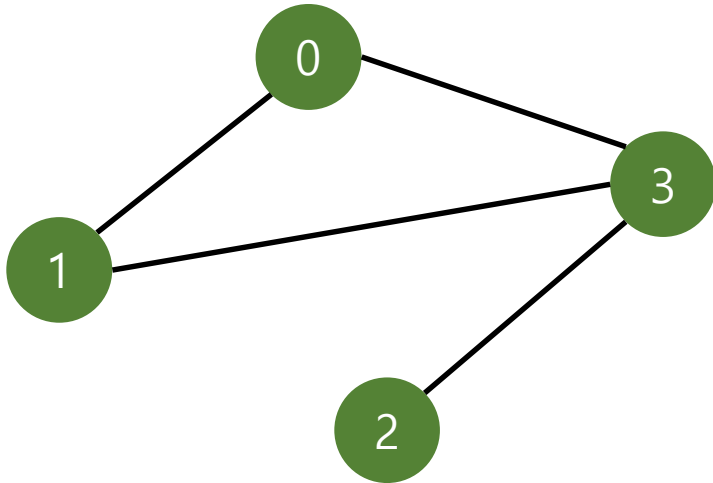
1. 인접하는 모든 노드 : $O(n)$
2. (u, v) 검사 : $O(1)$

	0	1	2	3
0	0	1	0	1
1	1	0	0	1
2	0	0	0	1
3	1	1	1	0

그래프 표현(graph representation)
: 인접 리스트(Adjacency list)

Operations

1. 인접하는 모든 노드 : $O(d(v))$
2. (u, v) 검사 : $O(d(v))$



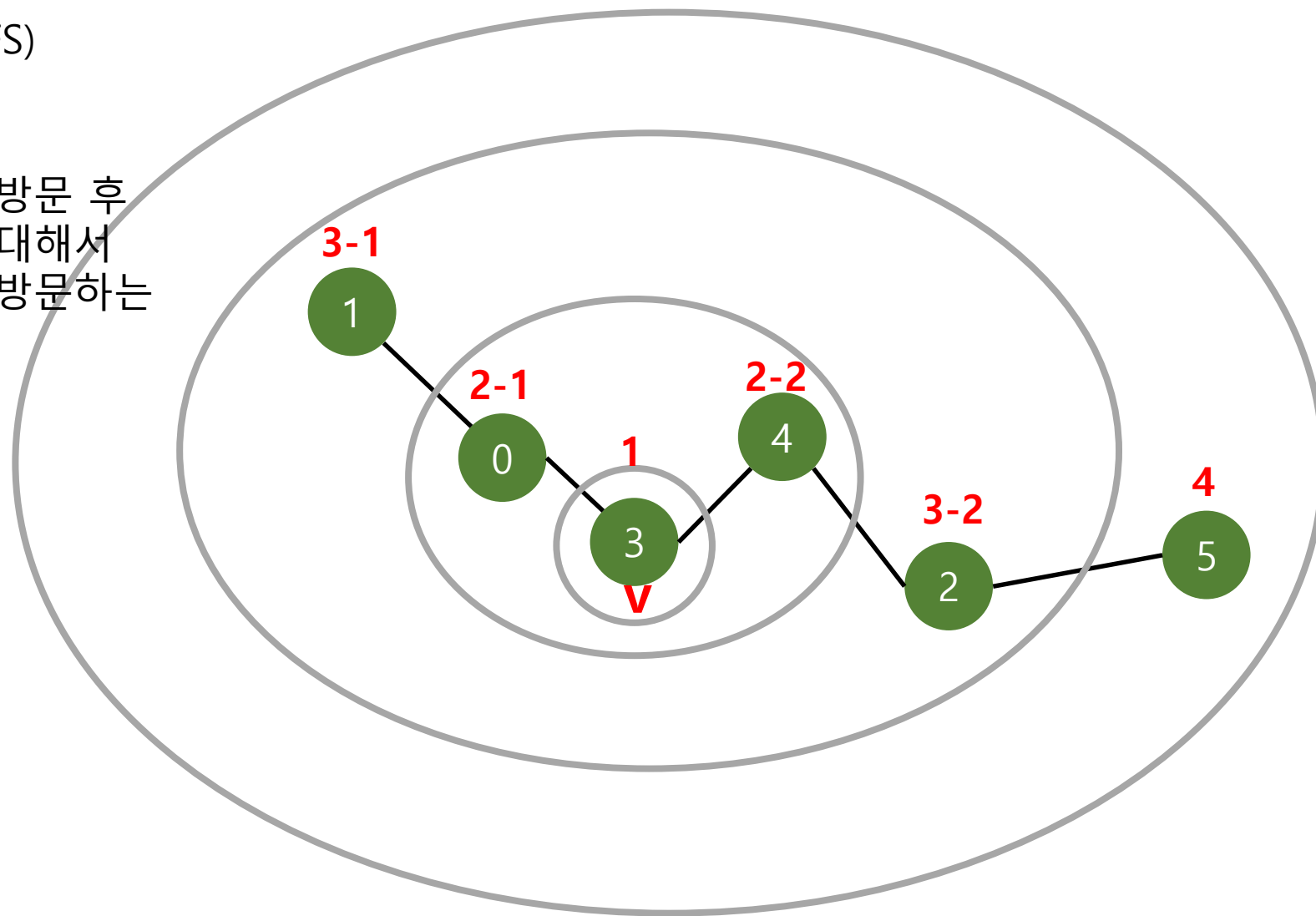
Graph traversal(그래프 순회)

그래프 G 에서 한 정점 v 가 주어졌을 때
모든 정점을 중복되지 않게 방문하는 것

1. 너비 우선 탐색(BFS, breath first search)
2. 깊이 우선 탐색(DFS, depth first search)

너비 우선 탐색(BFS)

정점 v 에서 시작하여
인접한 정점을 모두 방문 후
방문한 인접 정점에 대해서
모든 인접한 정점을 방문하는
방식



깊이 우선 탐색(DFS)

정점 v 에서 시작하여
방문하지 않은 정점 중
한 방향으로 쭉 따라간 후
방문할 정점이 없으면 다시 돌아와
방문하지 않은 다른 정점을 따라
다시 쭉 따라간다.
처음 시작한 정점에 돌아오고 더 이상
방문할 정점이 남아있지 않다면 종료

