

---

# 그래프

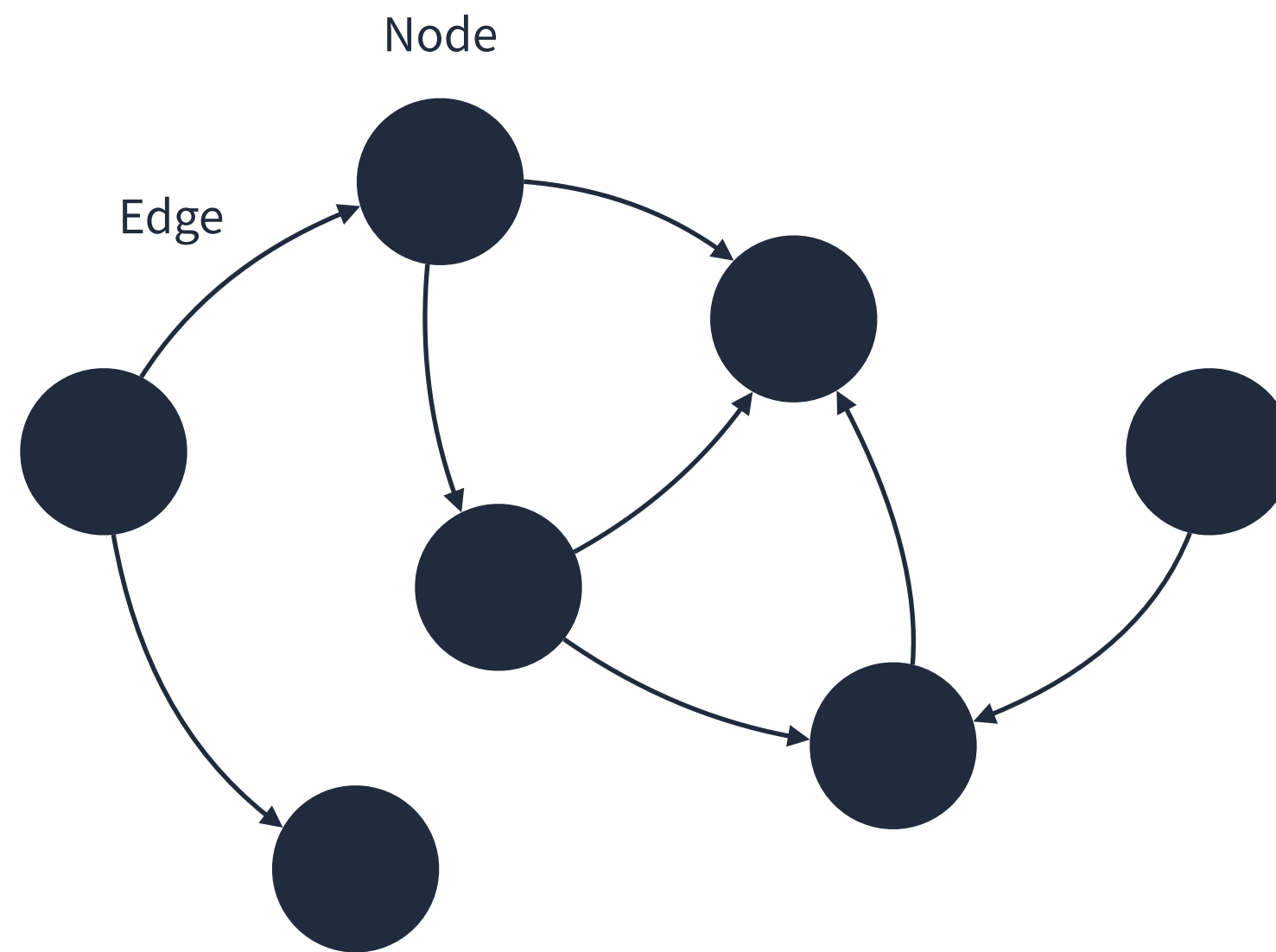
---

코딩테스트 광탈방지 A to Z : JavaScript - 이선희 @kciter

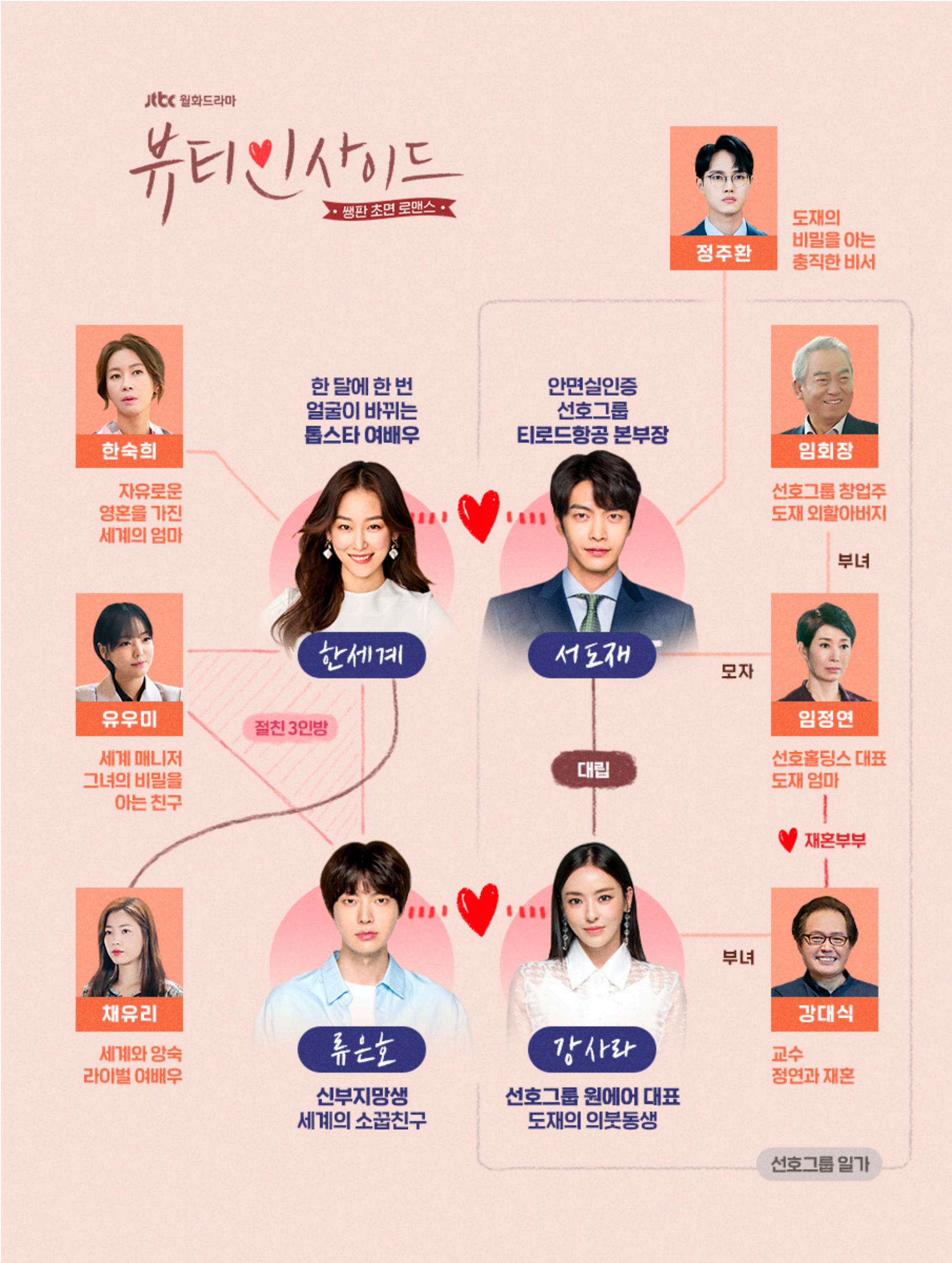
# JS

# 그래프

정점과 정점 사이를 연결하는 간선으로 이루어진 비선형 자료구조  
정점 집합과 간선 집합으로 표현할 수 있다.











페이지 랭크

전체

이미지

동영상

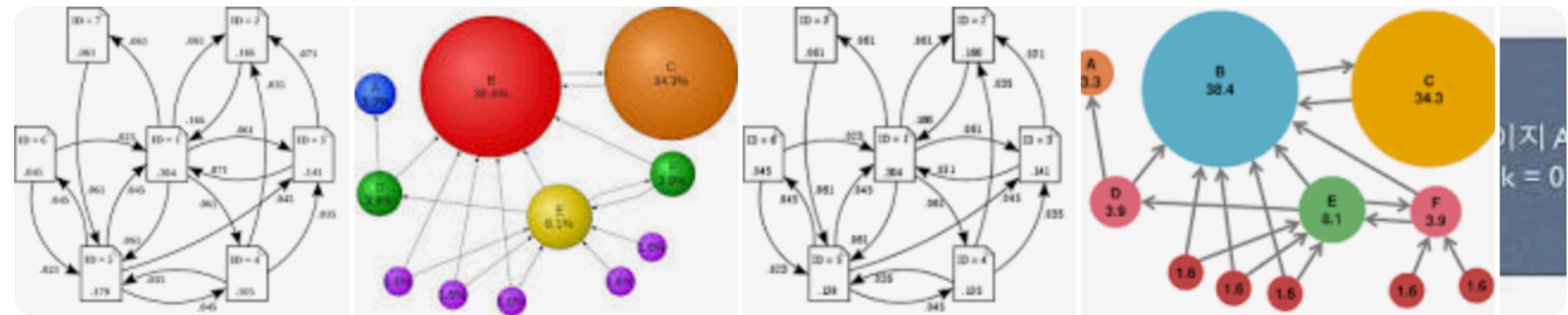
뉴스

지도

더보기

도구

검색결과 약 10,200,000개 (0.43초)



**페이지랭크**(PageRank)는 월드 와이드 웹과 같은 하이퍼링크 구조를 가지는 문서에 상대적 중요도에 따라 가중치를 부여하는 방법으로, 웹사이트 **페이지**의 중요도를 측정하기 위해 구글 검색에 쓰이는 알고리즘이다. 이 알고리즘은 서로간에 인용과 참조로 연결된 임의의 묶음에 적용할 수 있다.

<https://ko.wikipedia.org/wiki/페이지랭크>

**페이지랭크 - 위키백과, 우리 모두의 백과사전**

[추천 스니펫 정보](#) · [사용자 의견](#)

<https://icim.nims.re.kr/post/easyMath>

**구글의 페이지랭크(PageRank) | 알기 쉬운 산업수학 | 산업수학 ...**

2019. 12. 16. — '처음 몇 십 개'처럼 검색 결과를 나열하는 기준으로 그 웹페이지가 얼마나 중요한지를 나타내는 수치(지표)가 필요하다. PageRank는 **페이지**의 중요도 ...

<https://www.hani.co.kr/arti/science/technology>

**그 알고리즘이 너무 매력적이었기에 그들은 구글을 세웠다 ...**

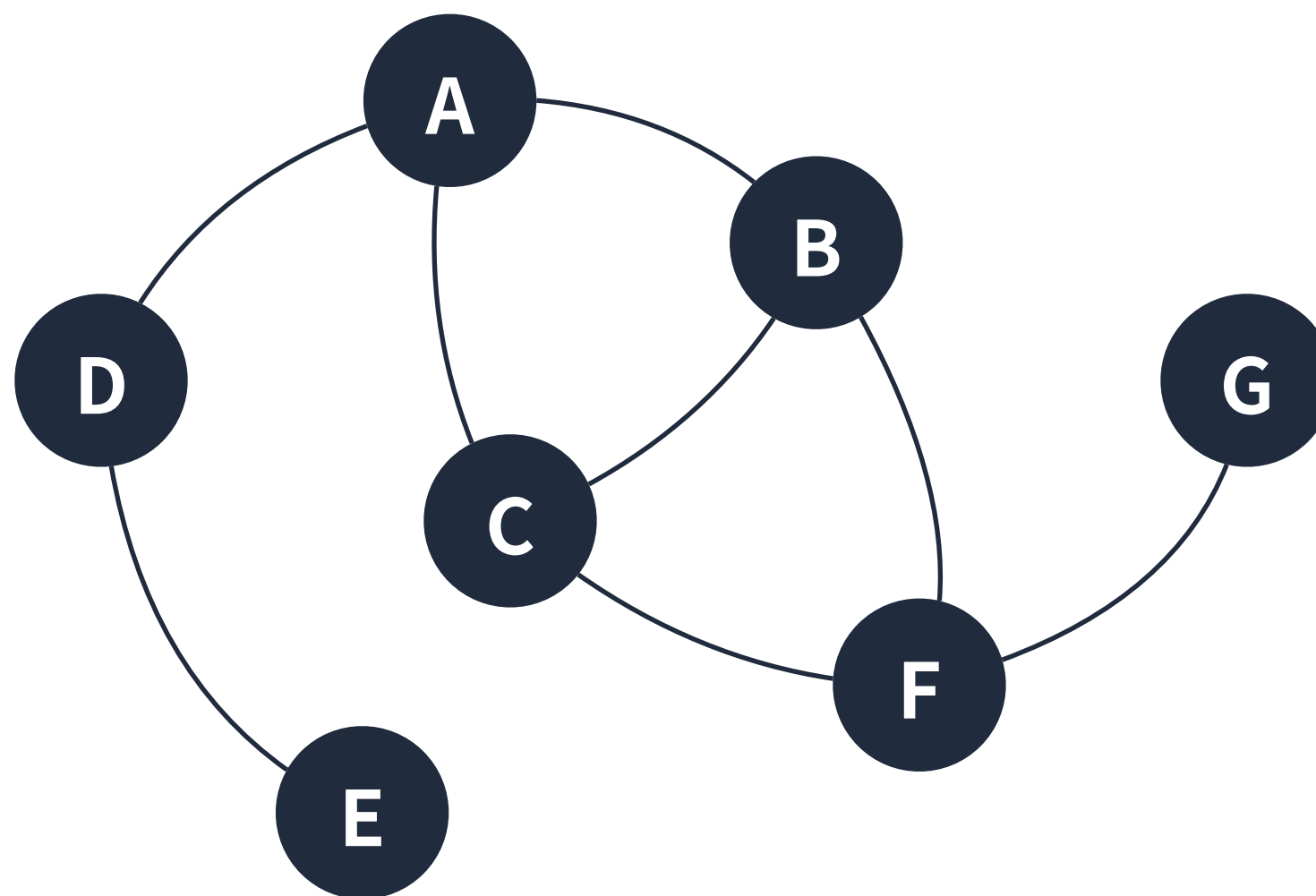
2019. 5. 17. — 검색 알고리즘 **페이지랭크**에 대한 이야기. **페이지랭크**. 출처: 위키미디어커먼스 Felipe Micaroni Lalli. 인터넷은 그 자체가 데이터다. 물론 인터넷상 ...

# 그래프의 특징

- 정점은 여러 개의 간선을 가질 수 있다.
- 크게 방향 그래프와 무방향 그래프로 나눌 수 있다.
- 간선은 가중치를 가질 수 있다.
- 사이클이 발생할 수 있다.

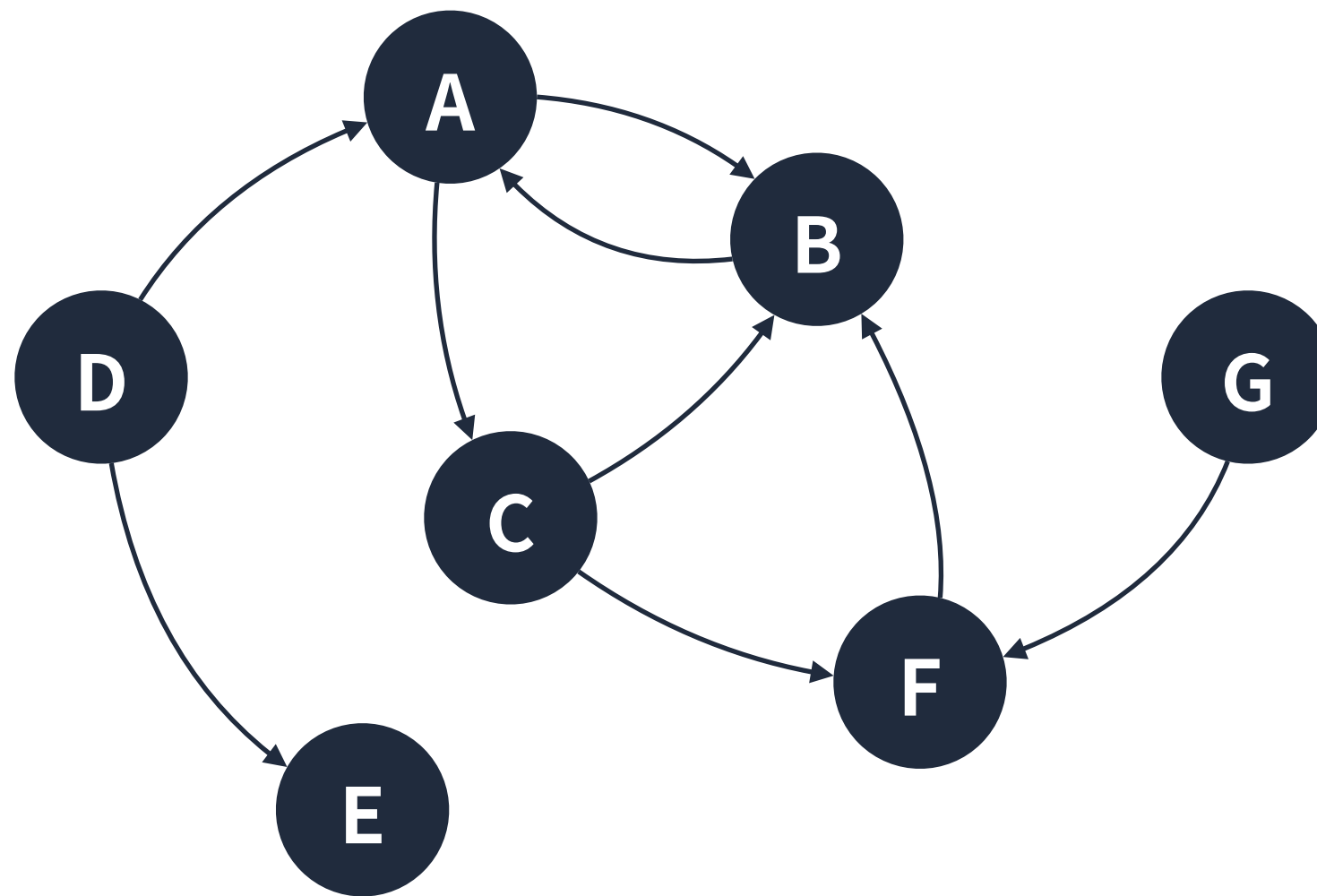
## 무방향 그래프

간선으로 이어진 정점끼리는 양방향으로 이동이 가능하다.  
표현하기에 (A, B)와 (B, A)는 같은 간선으로 취급된다.  
ex) 양방향 통행 도로



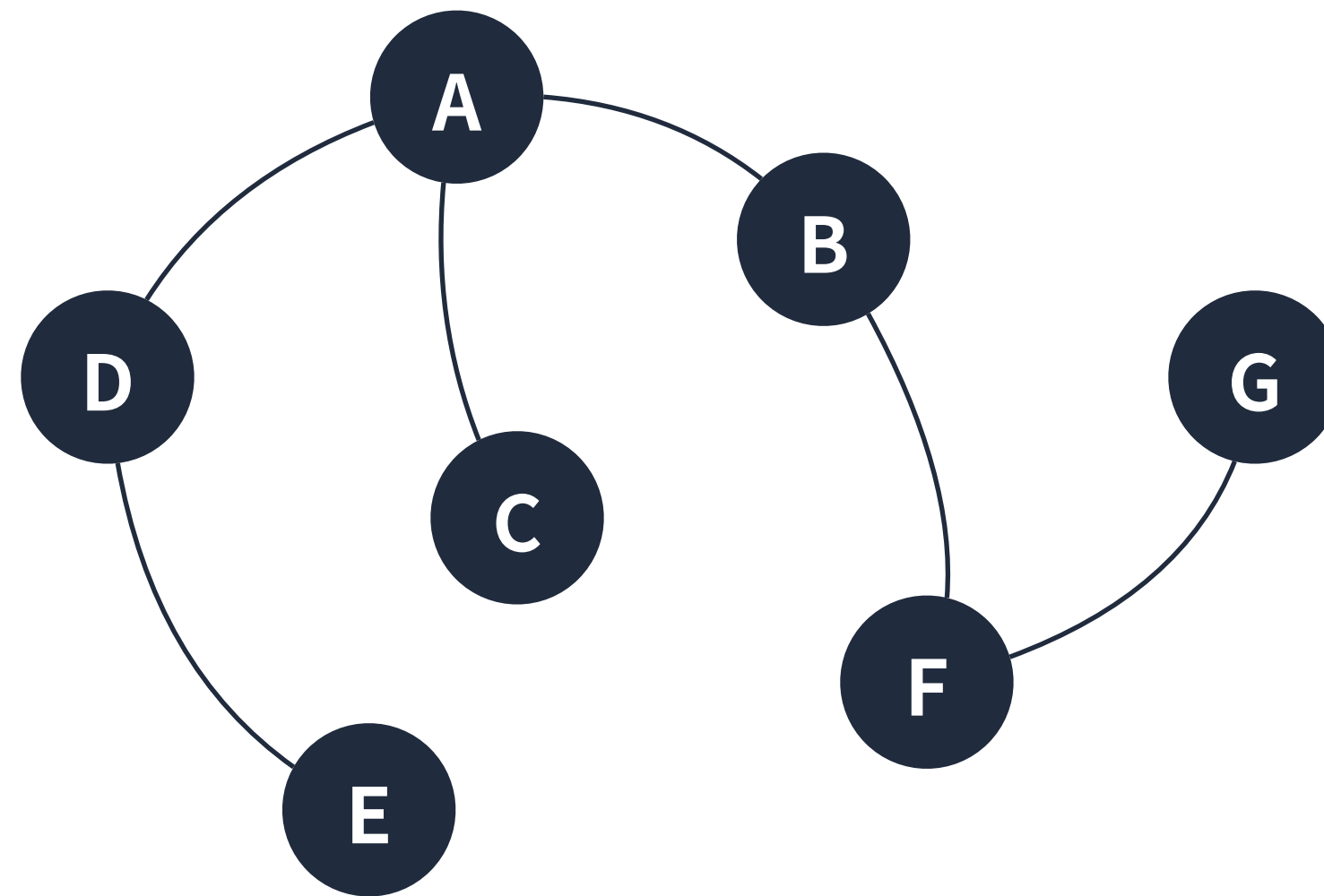
## 방향 그래프

간선에 방향성이 존재하는 그래프.  
양방향으로 갈 수 있더라도  $\langle A, B \rangle$ 와  $\langle B, A \rangle$ 는 다른 간선으로 취급된다.  
ex) 일방 통행



## 연결 그래프

모든 정점이 서로 이동 가능한 상태인 그래프

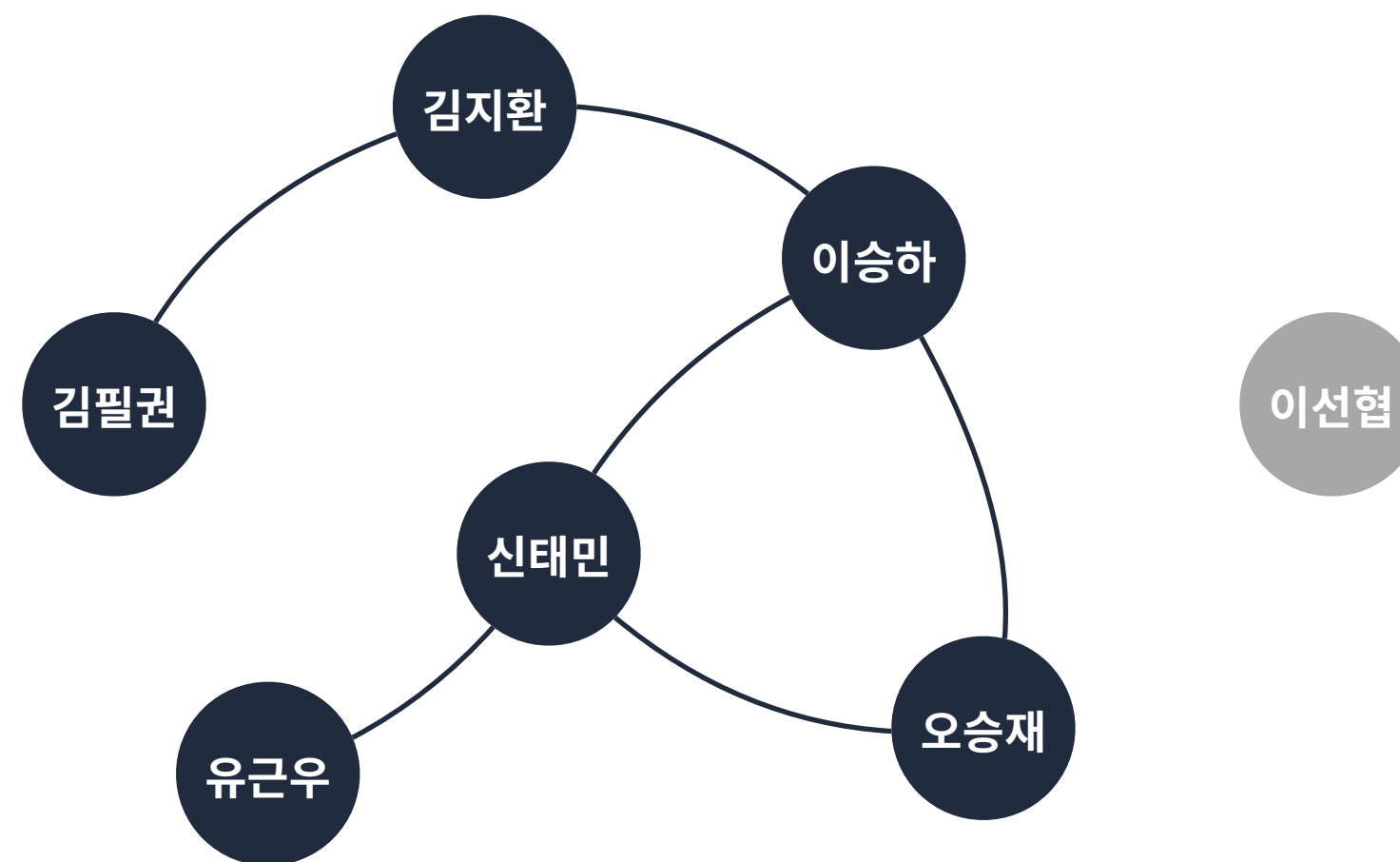




# 비연결 그래프

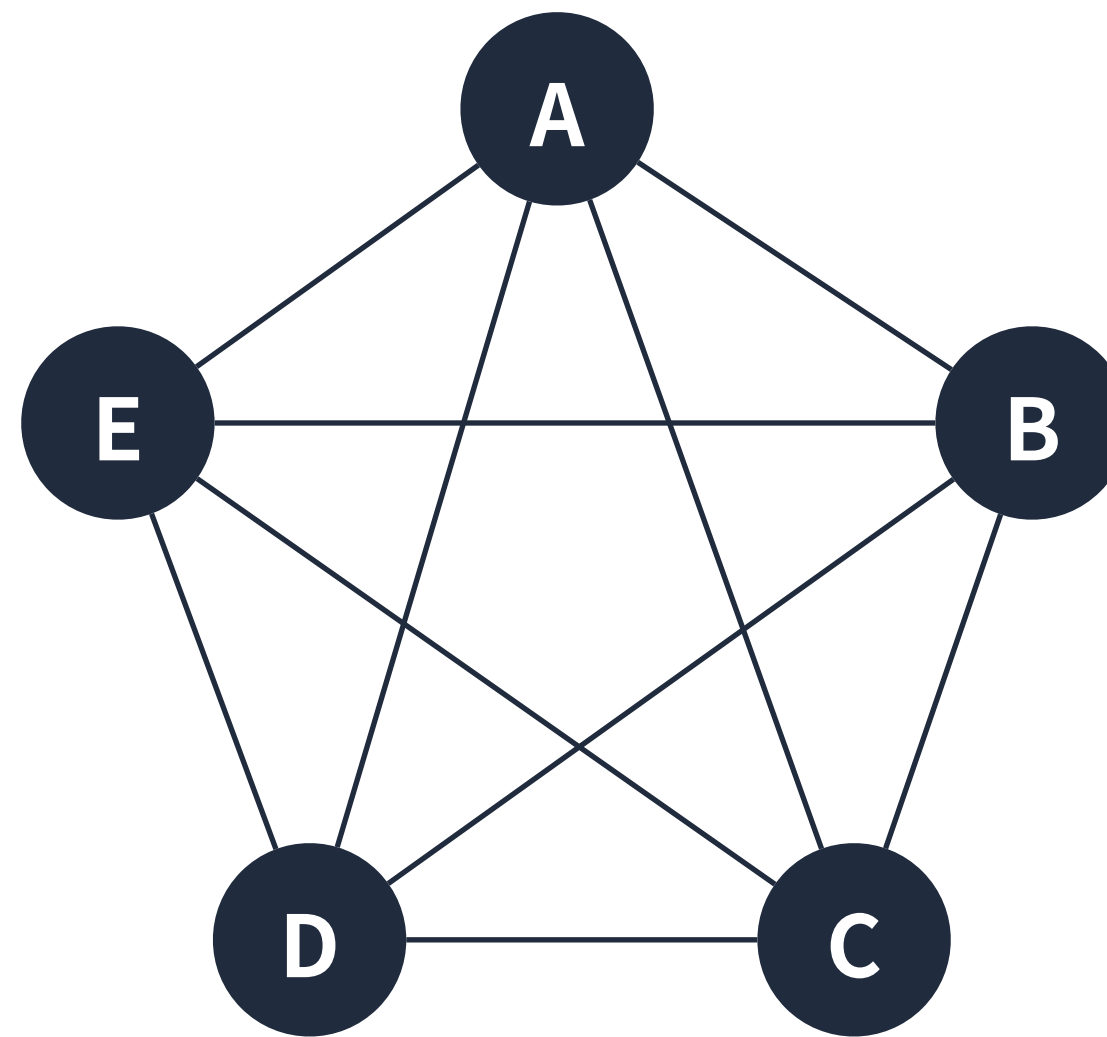
특정 정점쌍 사이에 간선이 존재하지 않는 그래프

ex) 친한 친구 설문 그래프



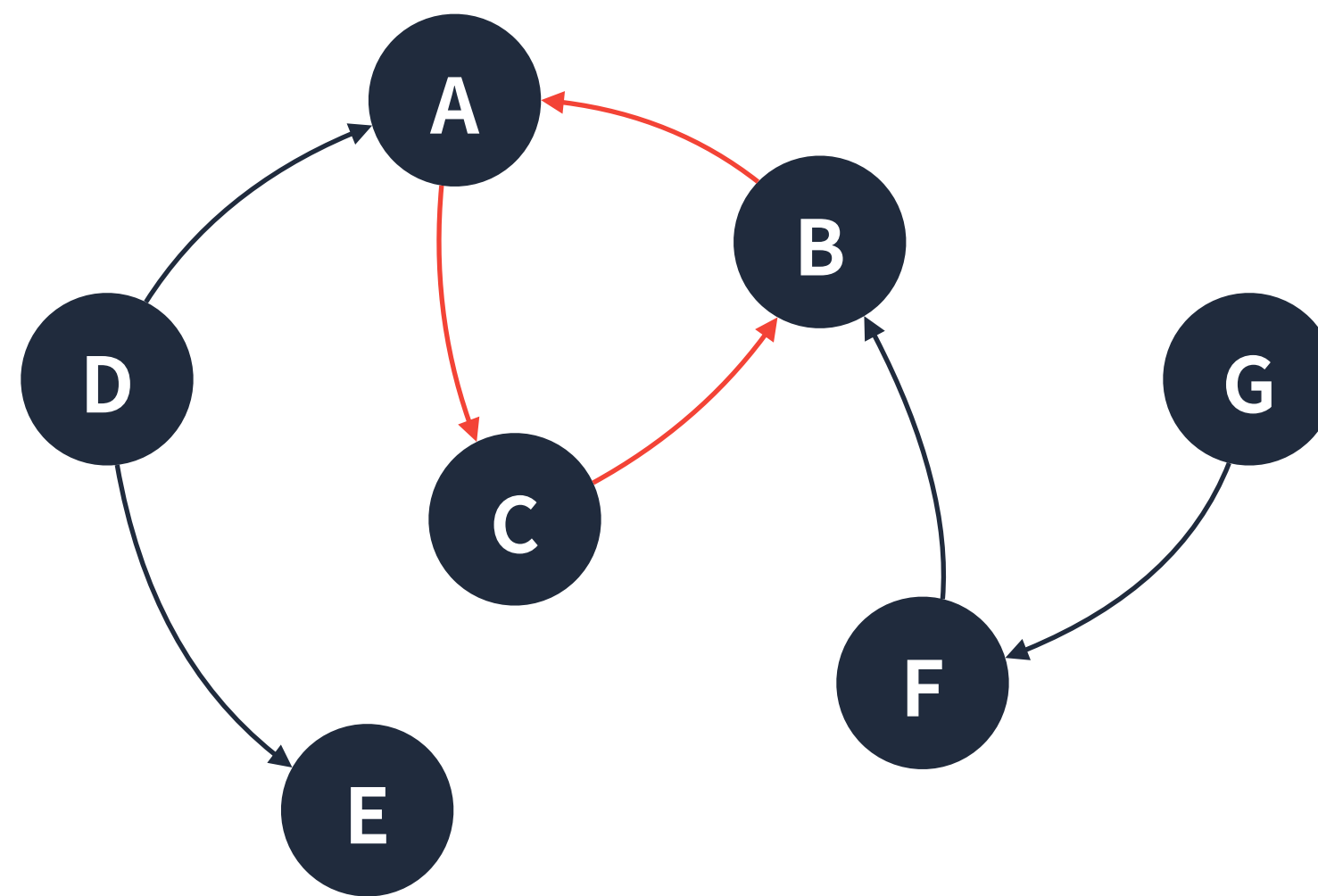
# 완전 그래프

모든 정점끼리 연결된 상태인 그래프



# 사이클

그래프의 정점과 간선의 부분 집합에서 순환이 되는 부분

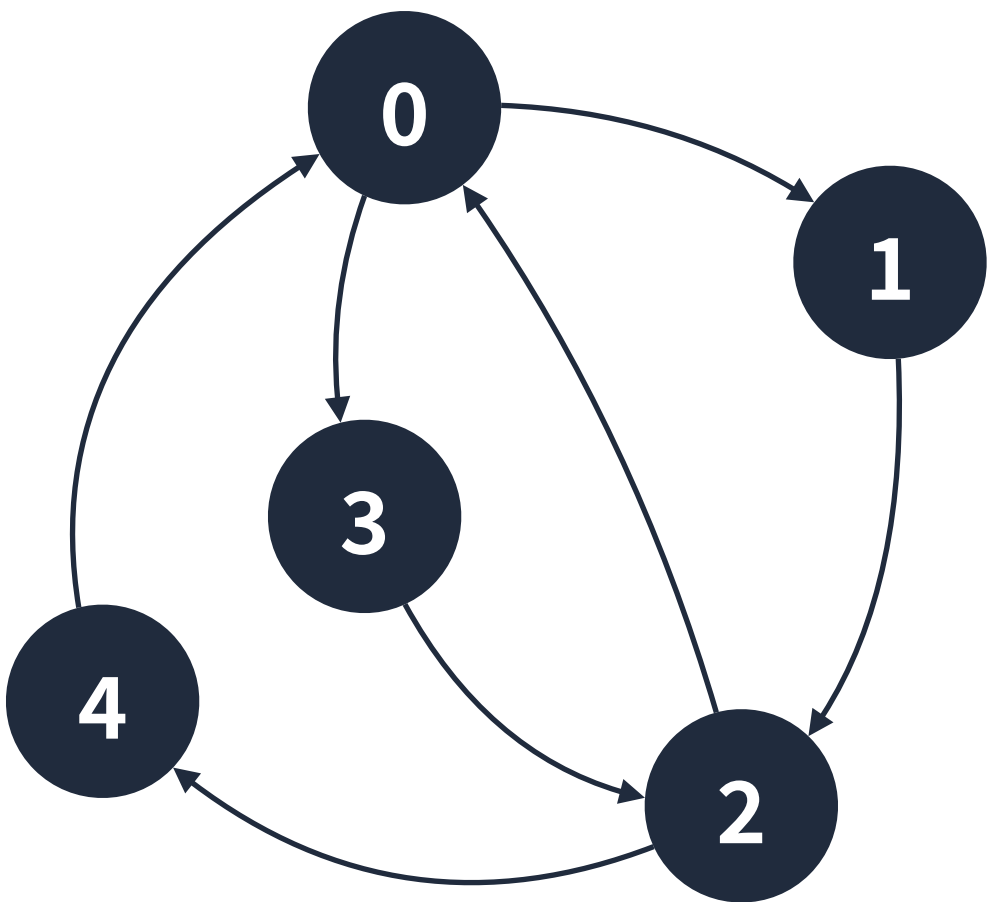


# 그래프의 구현 방법

인접 행렬, 인접 리스트 두 가지 방식으로 그래프를 표현할 수 있다.

Adjacency Matrix

0	1	0	1	0
0	0	1	0	0
1	0	0	0	1
0	0	1	0	0
1	0	0	0	0



Adjacency List

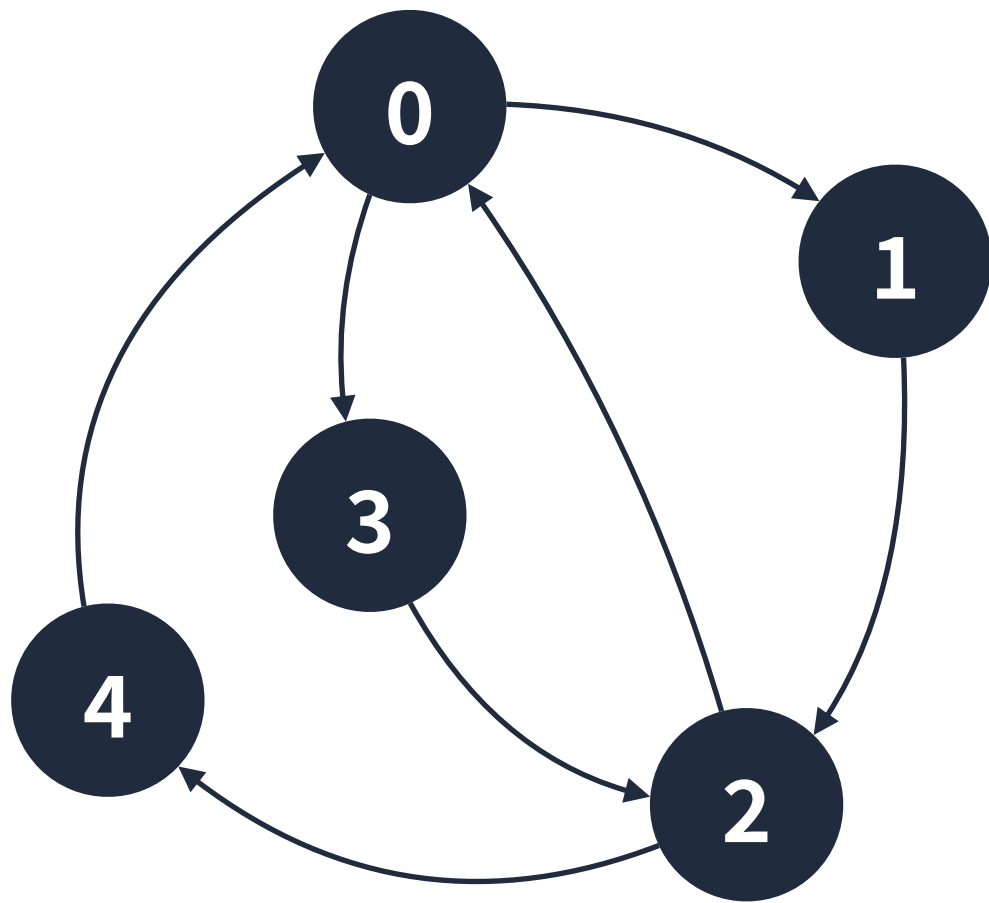
0		1	3
1		2	
2		0	4
3		2	
4		0	



# JavaScript에서 사용법

# 인접 행렬

Graph



Code

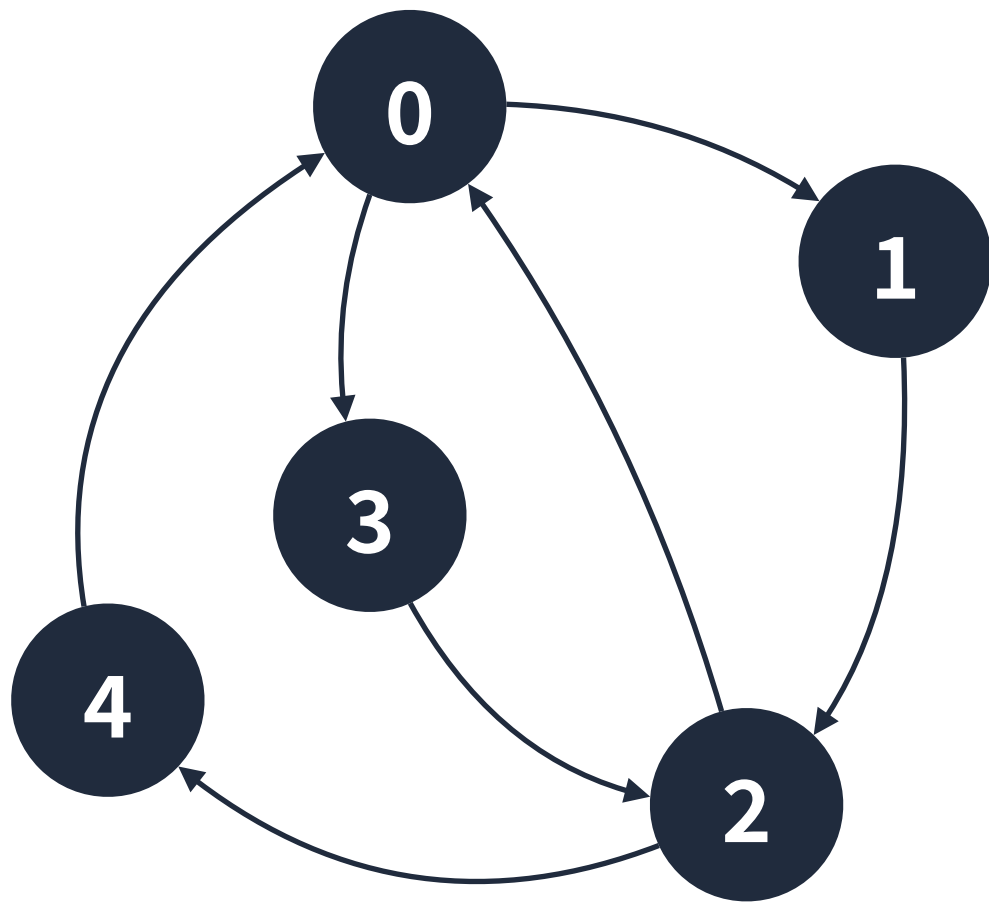
```
const graph = Array.from(
  Array(5),
  () => Array(5).fill(false)
);
graph[0][1] = true; // 0 -> 1
graph[0][3] = true; // 0 -> 3
graph[1][2] = true; // 1 -> 2
graph[2][0] = true; // 2 -> 0
graph[2][4] = true; // 2 -> 4
graph[3][2] = true; // 3 -> 2
graph[4][0] = true; // 4 -> 0
```

Result

0	1	0	1	0
0	0	1	0	0
1	0	0	0	1
0	0	1	0	0
1	0	0	0	0

# 인접 리스트

Graph



Code

```
const graph = Array.from(
  Array(5),
  () => []
);
graph[0].push(1); // 0 -> 1
graph[0].push(3); // 0 -> 3
graph[1].push(2); // 1 -> 2
graph[2].push(0); // 2 -> 0
graph[2].push(4); // 2 -> 4
graph[3].push(2); // 3 -> 2
graph[4].push(0); // 4 -> 0
```

Result

0		1	3
1		2	
2		0	4
3		2	
4		0	

---

# 그래프

---

코딩테스트 광탈방지 A to Z : JavaScript - 이선희 @kciter

# JS