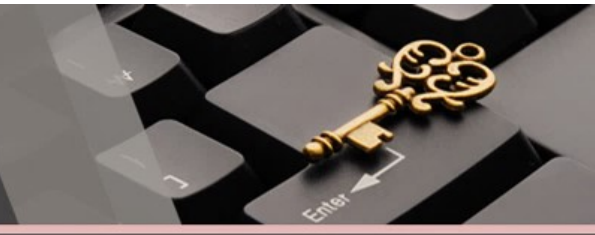
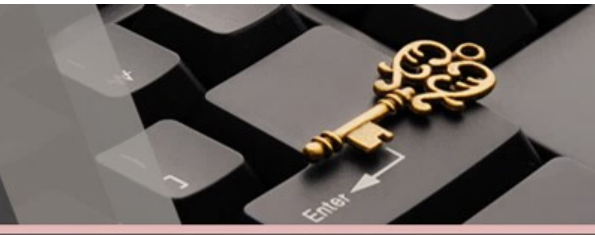
실전 알고리즘 0x11강
위상 정렬

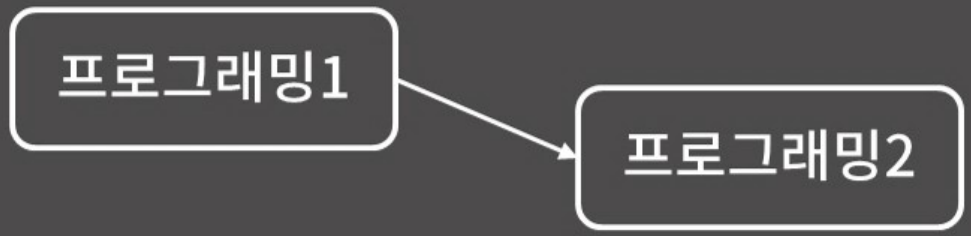
BaaaaaaaaaaaaaaaarkingDog

목차

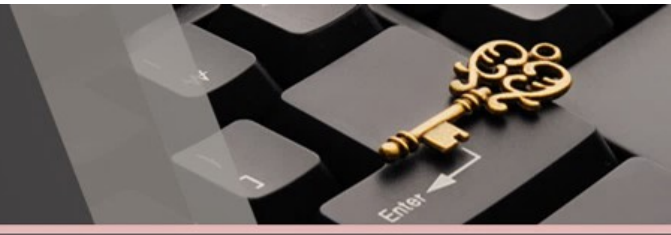


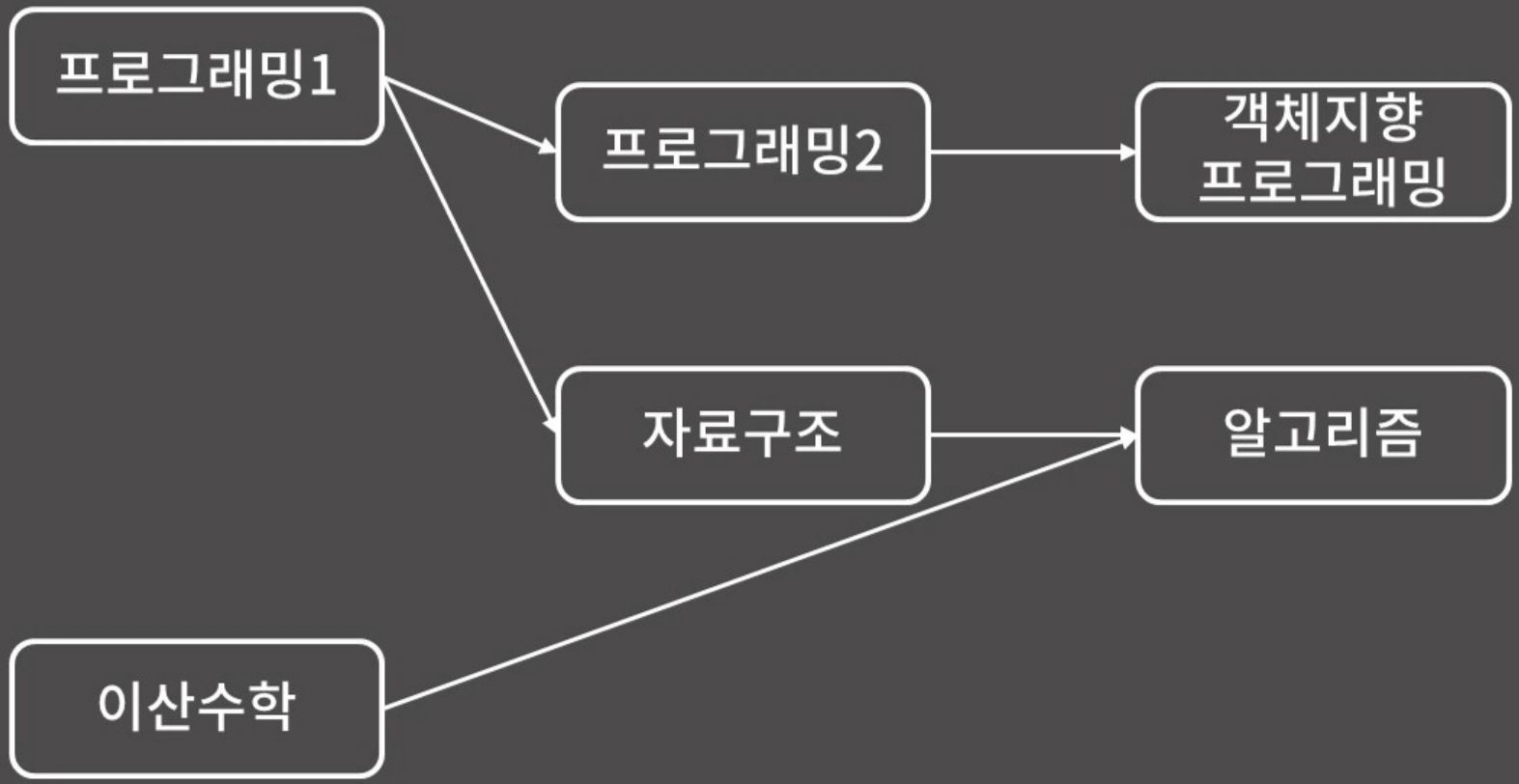
0x00 위상 정렬의 정의



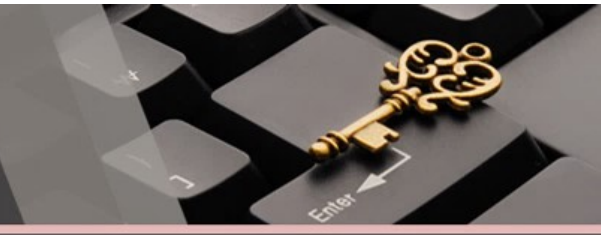


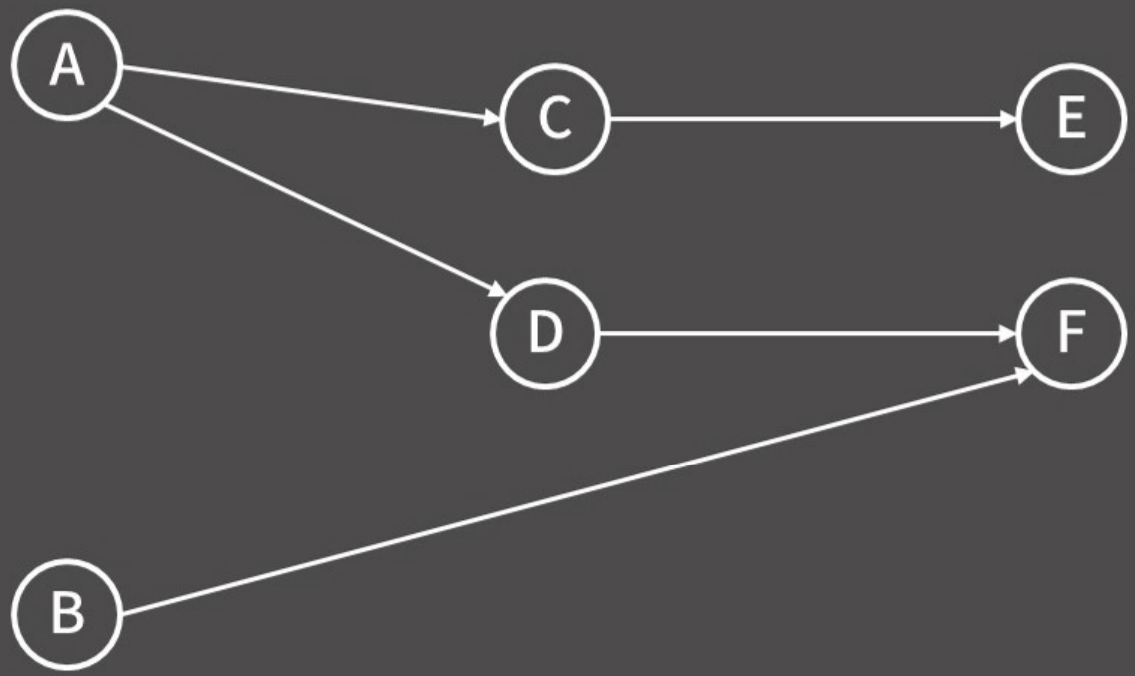
0x00 위상 정렬의 정의





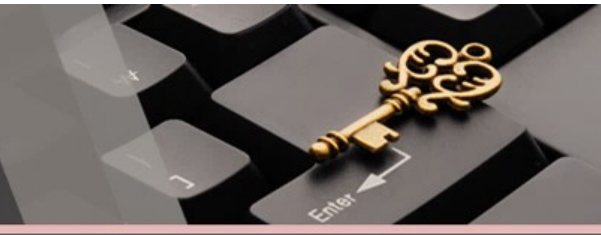
0x00 위상 정렬의 정의

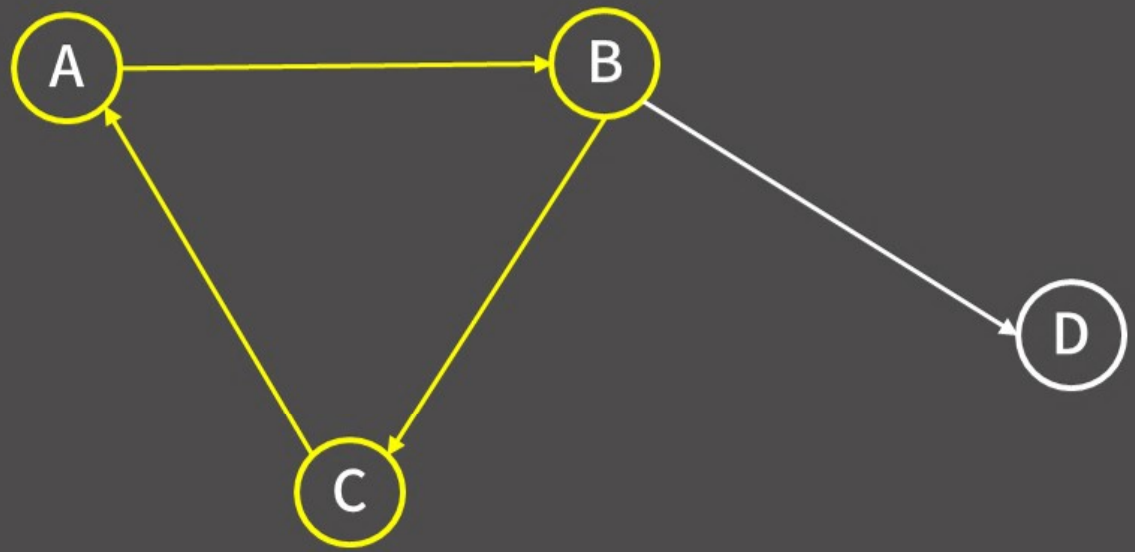




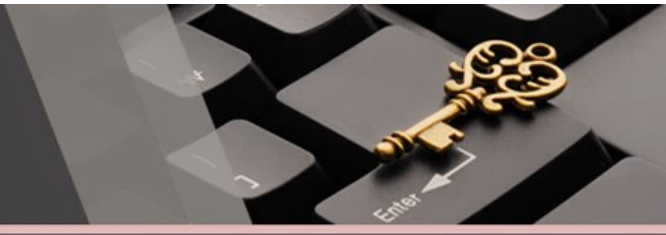
위상 정렬(Topological Sort) : 방향 그래프에서 간선으로 주어진 정점 간 선후관계를 위배하지 않도록 나열하는 정렬

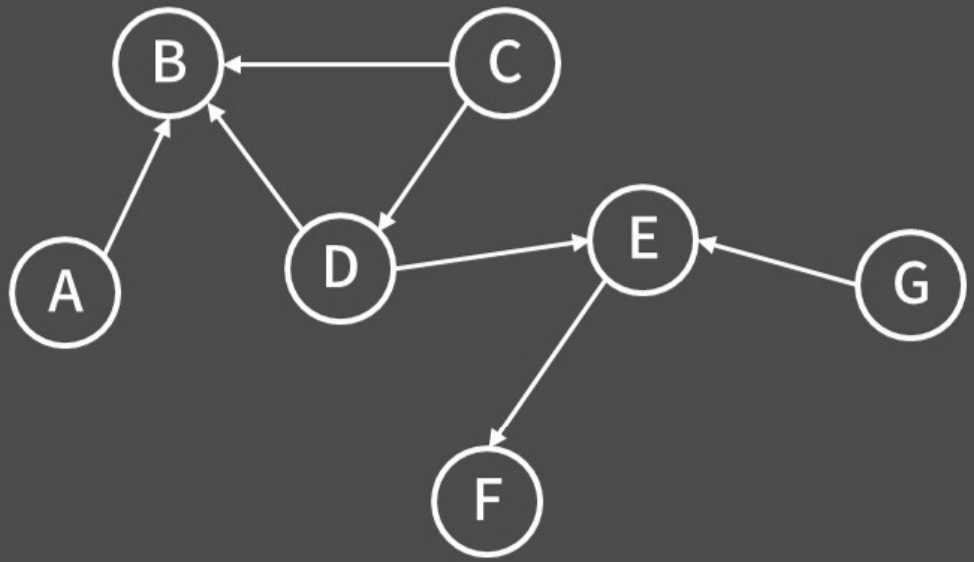
0x00 위상 정렬의 정의



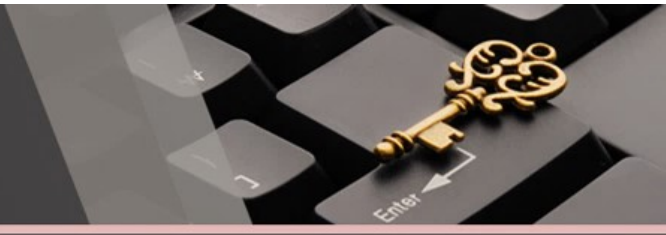


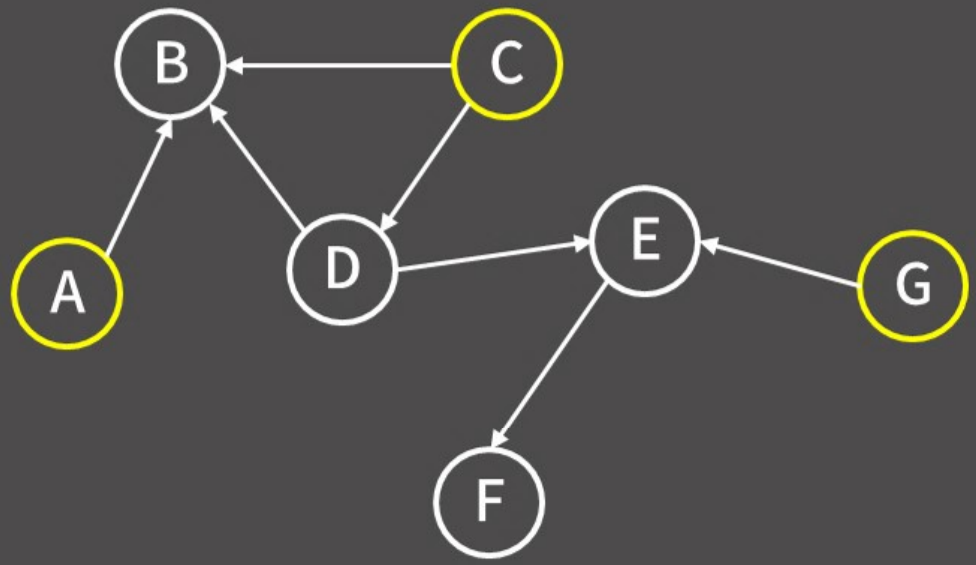
0x01 위상 정렬 알고리즘



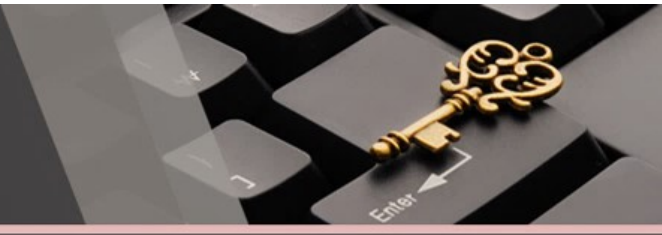


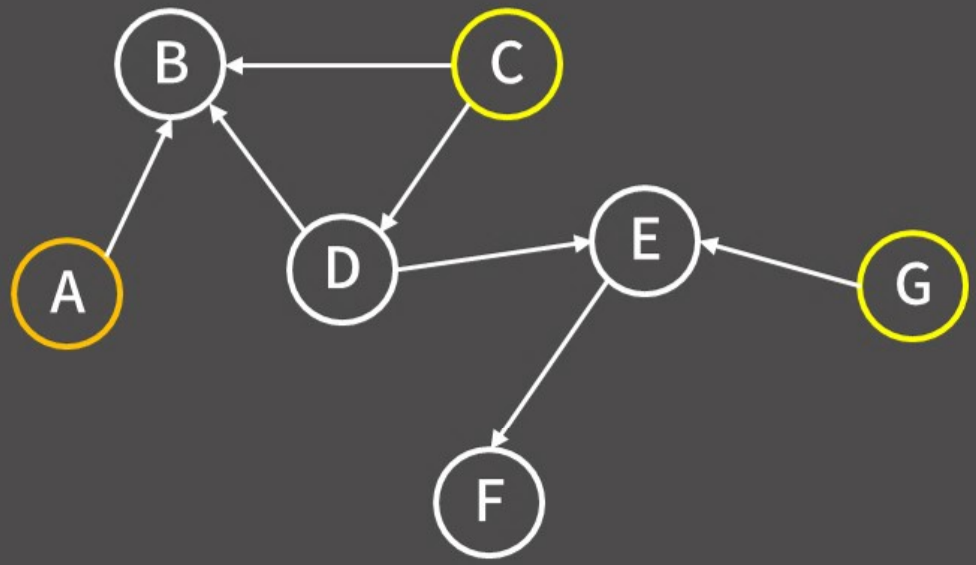
0x01 위상 정렬 알고리즘





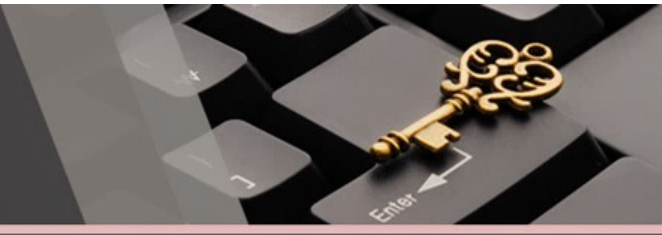
0x01 위상 정렬 알고리즘

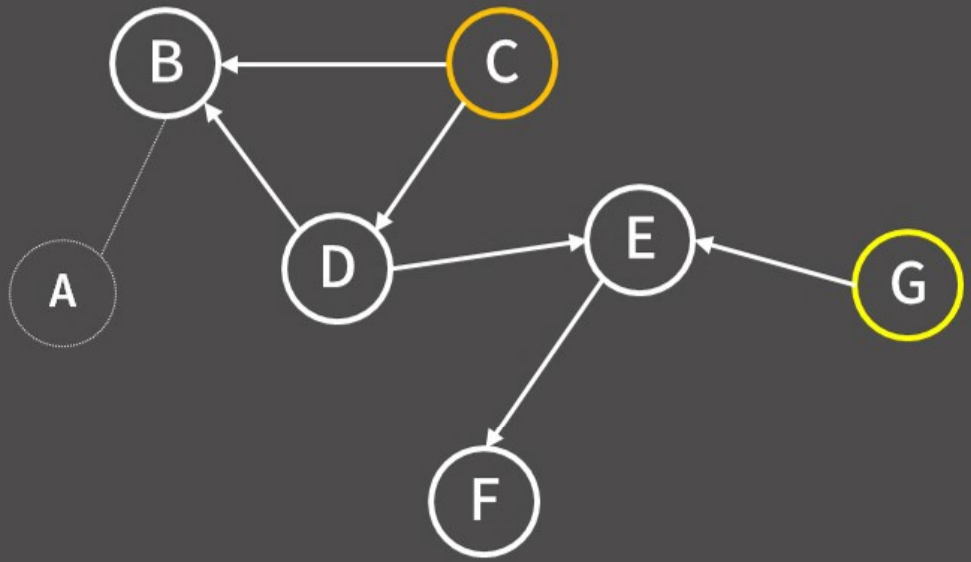




위상 정렬 결과

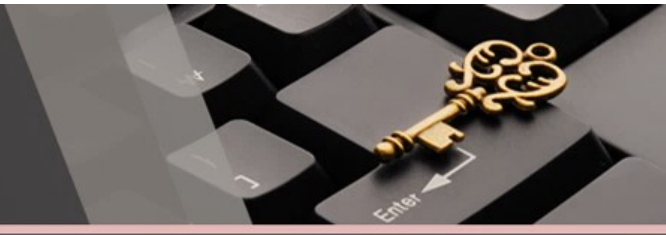
0x01 위상 정렬 알고리즘

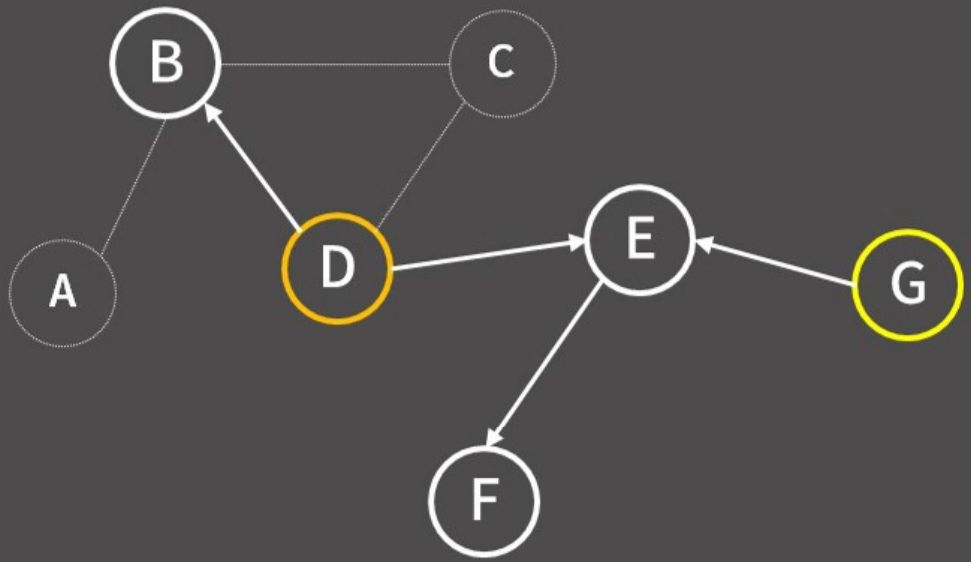




위상 정렬 결과

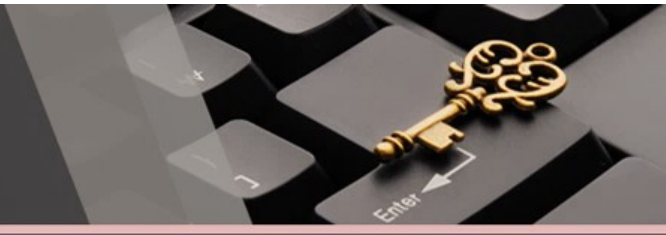
0x01 위상 정렬 알고리즘

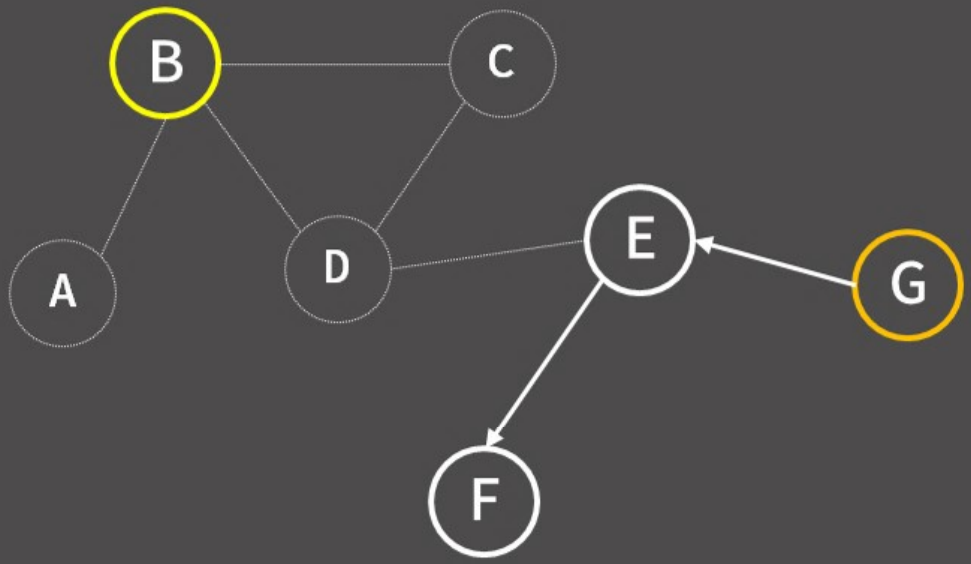




위상 정렬 결과 A C D

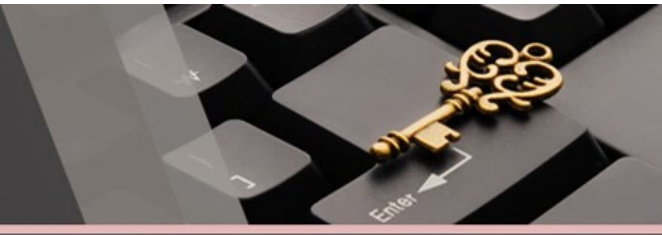
0x01 위상 정렬 알고리즘

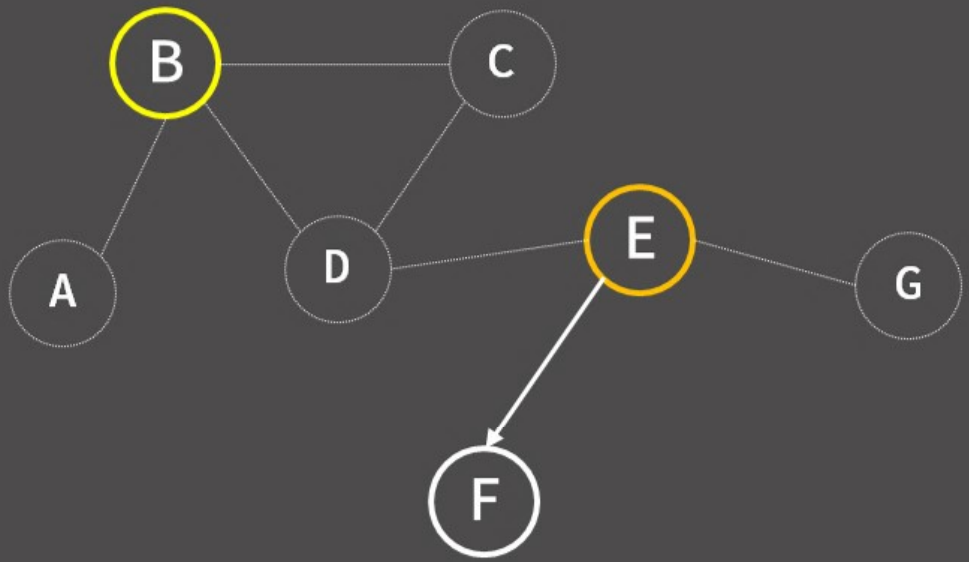




위상 정렬 결과

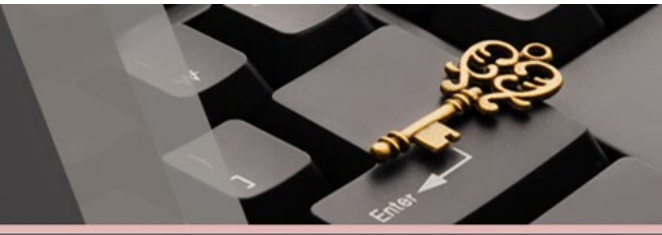
0x01 위상 정렬 알고리즘

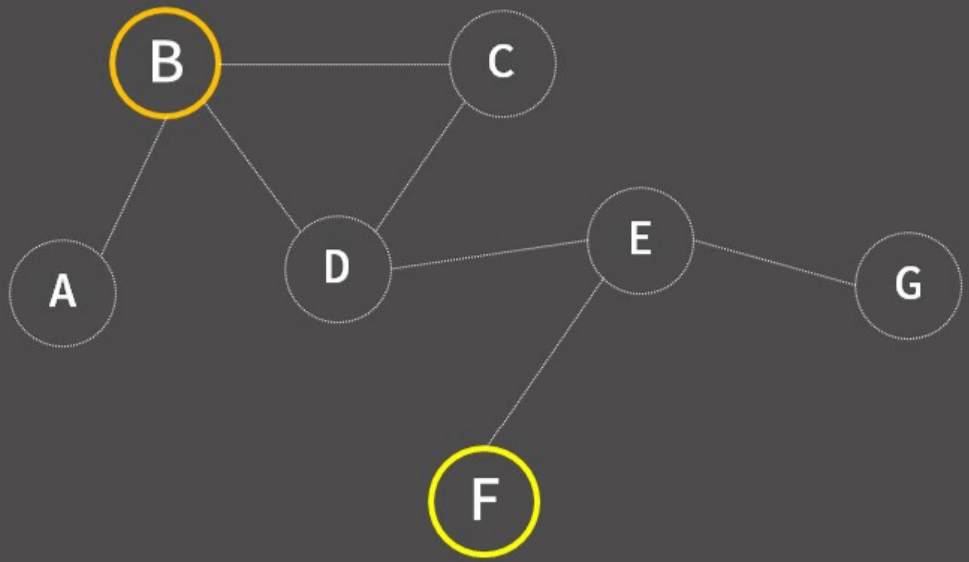




위상 정렬 결과

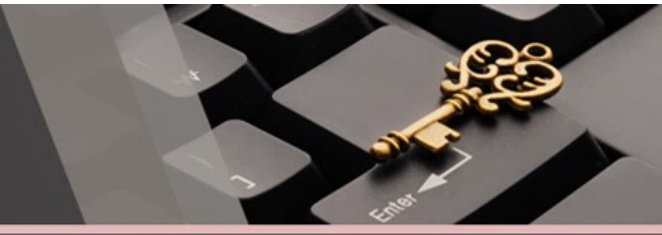
0x01 위상 정렬 알고리즘

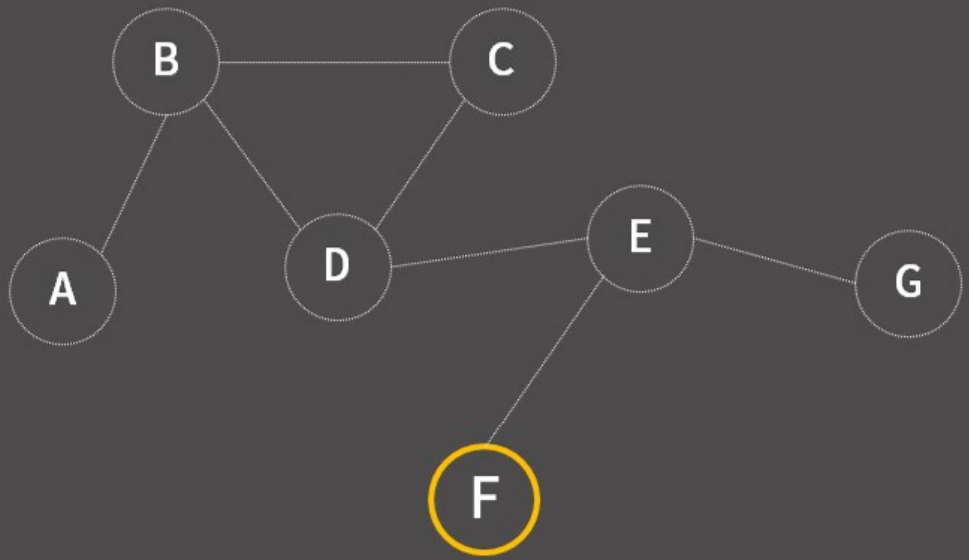




위상 정렬 결과

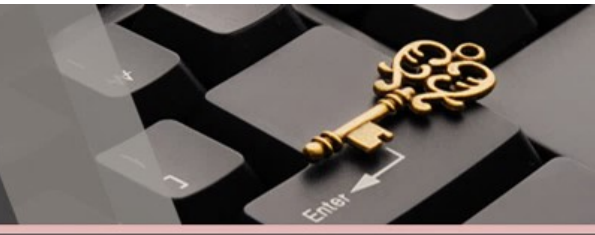
0x01 위상 정렬 알고리즘

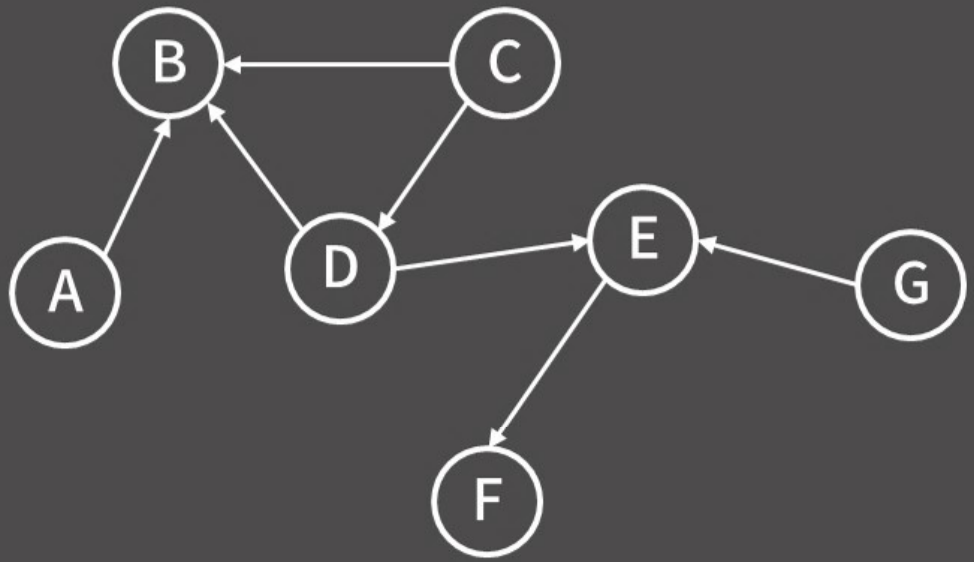




위상 정렬 결과

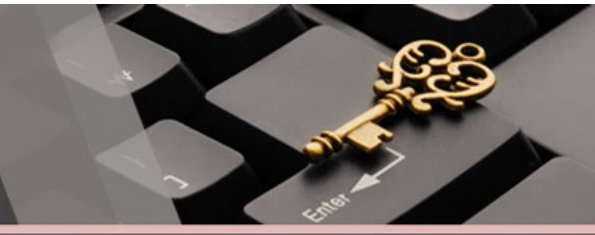
0x01 위상 정렬 알고리즘





위상 정렬 결과

0x02 위상 정렬 구현

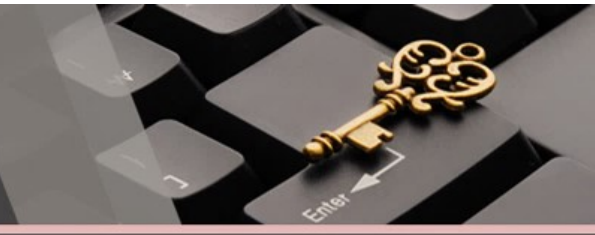


구현의 편의를 위한 성질

1. 정점과 간선을 실제로 지울 필요 없이 미리 indegree의 갯수를 저장해두었다가 매번 뻗어 나가는 정점들의 indegree 값만 1 감소시켜도 과정을 수행할 수 있다.

2. indegree가 0인 정점을 구하기 위해 매번 모든 정점들을 다 확인하는 대신 목록을 따로 저장하고 있다가 직전에 제거한 정점에서 연결된 정점들만 추가하면 된다.

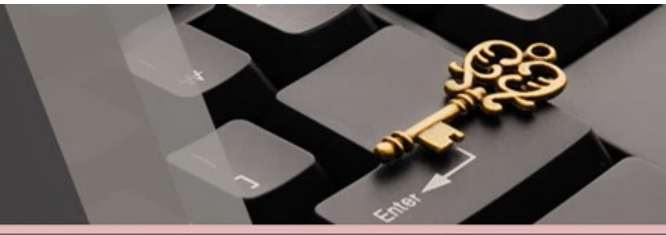
0x02 위상 정렬 구현

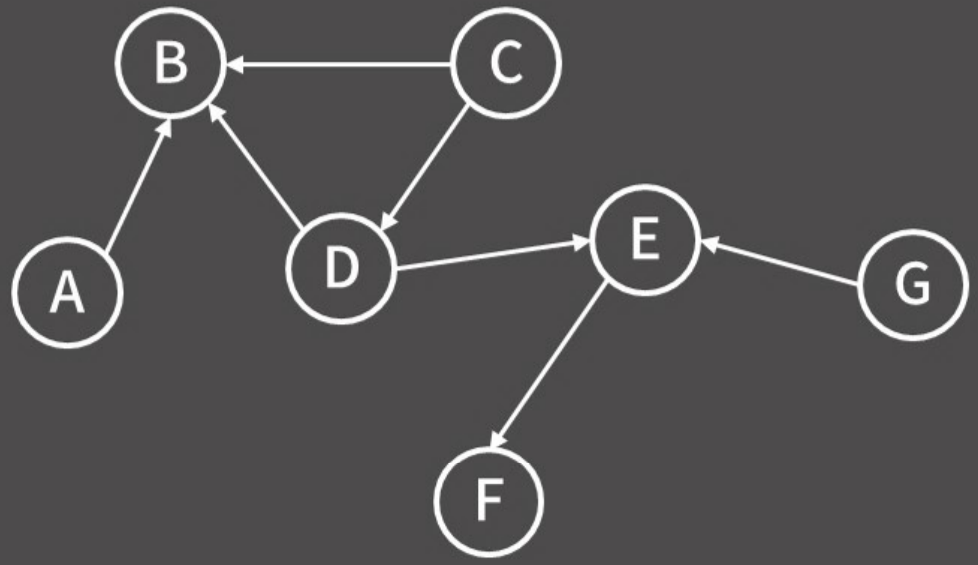


위상정렬 알고리즘

1. 맨 처음 모든 간선을 읽으며 Indegree 테이블을 채운다.
2. Indegree가 0인 정점들을 모두 큐에 넣는다.
3. 큐의 front에 있는 정점을 가져와 위상 정렬 결과에 추가한다.
4. 해당 정점으로부터 연결된 모든 정점의 Indegree 값을 1 감소시킨다. 이 때 Indegree가 0이 되었다면 그 정점을 큐에 추가한다.
5. 큐가 빌 때 까지 3, 4번 과정을 반복한다.

0x02 위상 정렬 구현





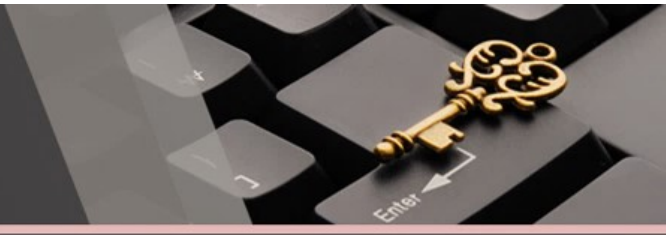
Indegree 배열

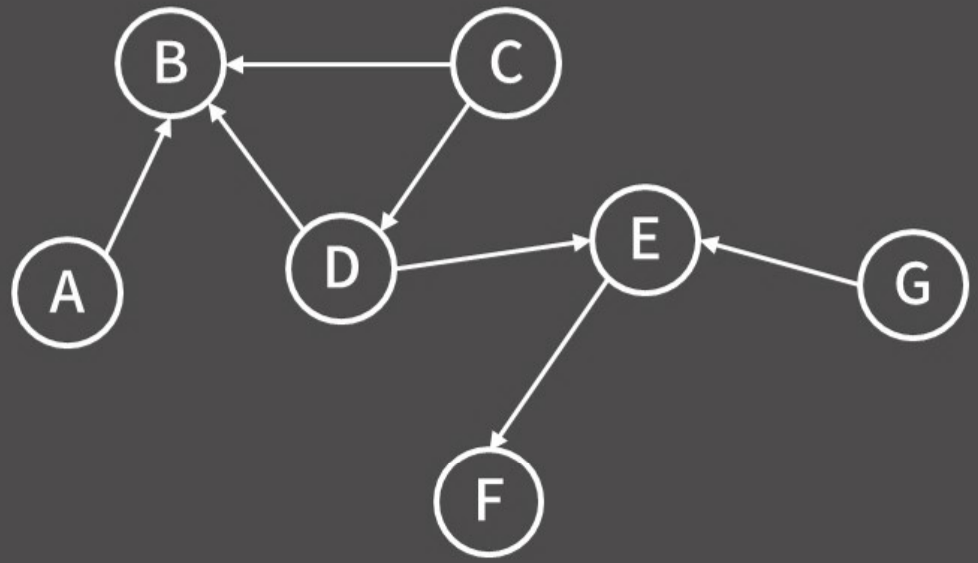
A	B	C	D	E	F	G
0	3	0	1	2	1	0

Indegree가 0인 정점을 저장할 큐

위상 정렬 결과

0x02 위상 정렬 구현





Indegree 배열

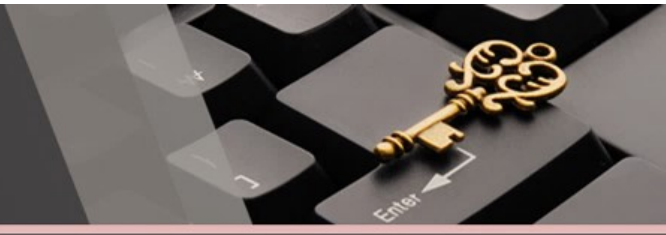
A	B	C	D	E	F	G
0	3	0	1	2	1	0

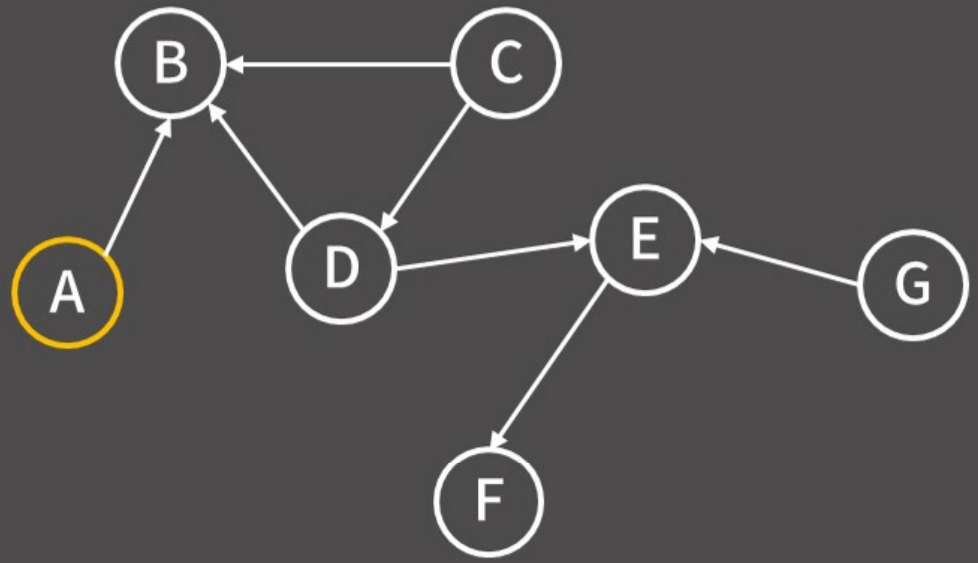
Indegree가 0인 정점을 저장할 큐

A C G

위상 정렬 결과

0x02 위상 정렬 구현





Indegree 배열

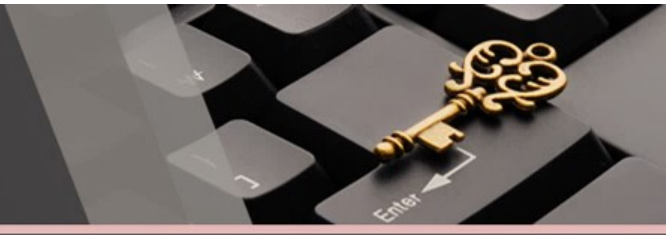
A	B	C	D	E	F	G
0	2	0	1	2	1	0

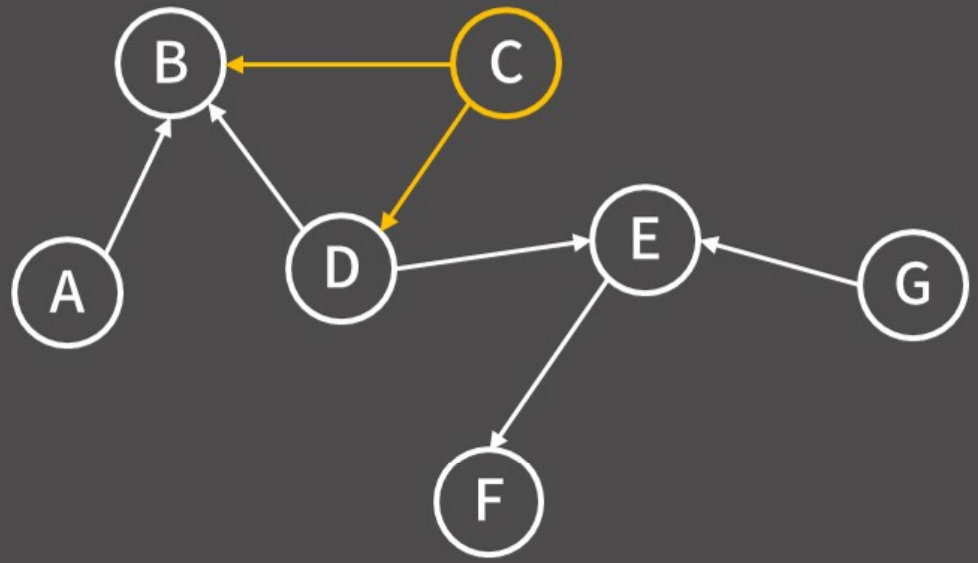
Indegree가 0인 정점을 저장할 큐

C G

위상 정렬 결과 A

0x02 위상 정렬 구현





Indegree 배열

A	B	C	D	E	F	G
0	1	0	0	2	1	0

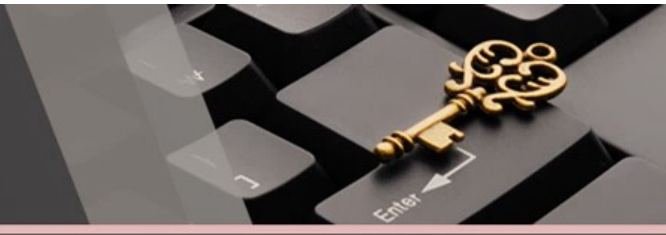
Indegree가 0인 정점을 저장할 큐

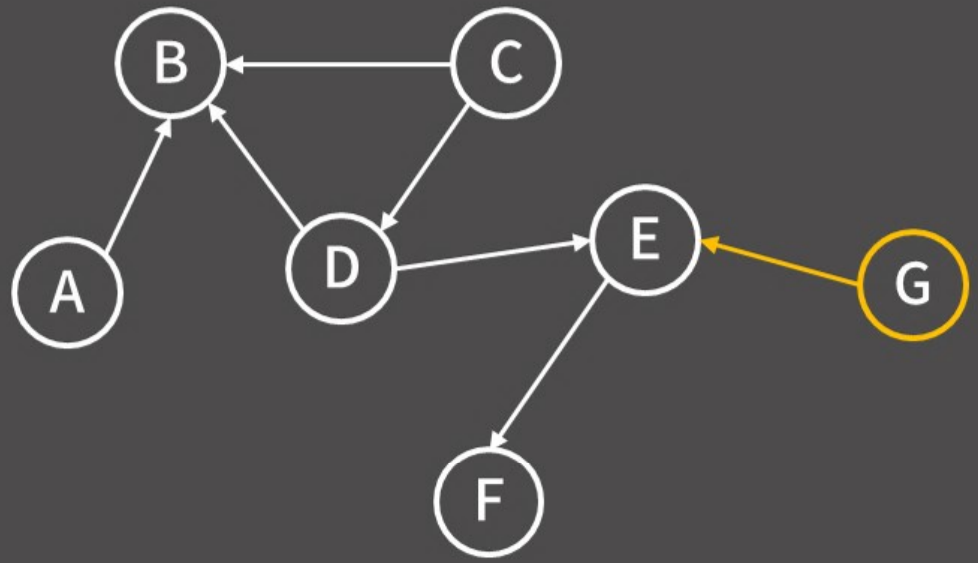
G D

위상 정렬 결과

A C

0x02 위상 정렬 구현





Indegree 배열

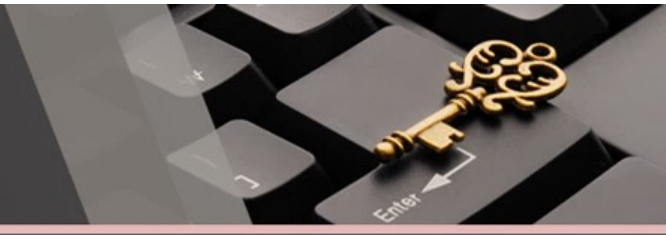
A	B	C	D	E	F	G
0	1	0	0	1	1	0

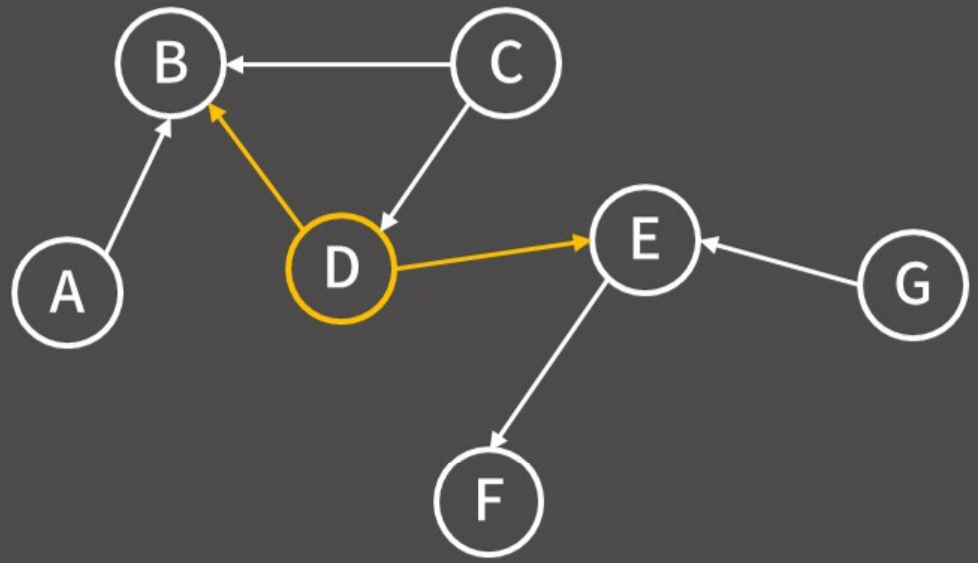
Indegree가 0인 정점을 저장할 큐

D

위상 정렬 결과 A C G

0x02 위상 정렬 구현





Indegree 배열

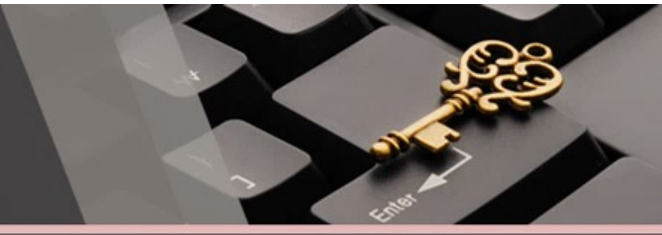
A	B	C	D	E	F	G
0	0	0	0	0	1	0

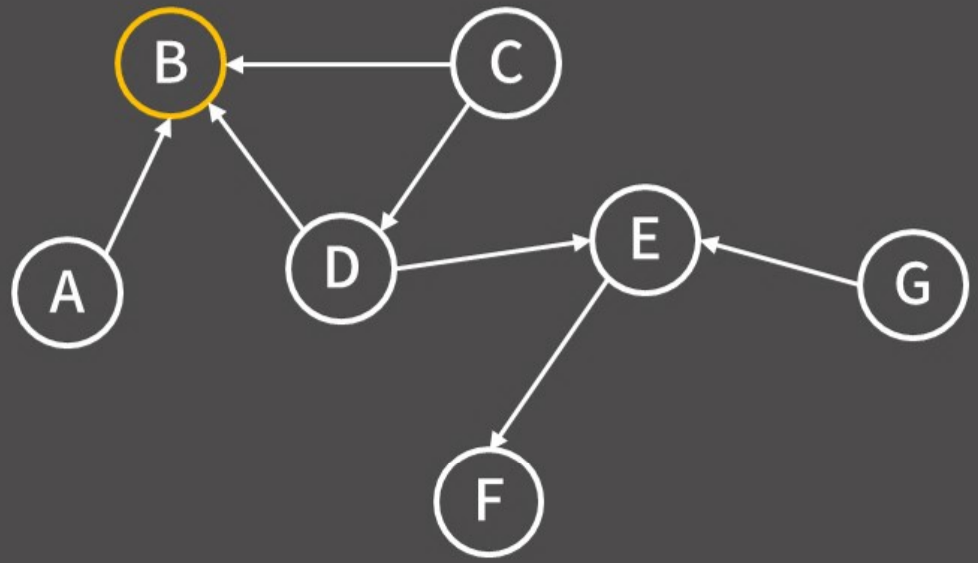
Indegree가 0인 정점을 저장할 큐

B E

위상 정렬 결과 A C G D

0x02 위상 정렬 구현





Indegree 배열

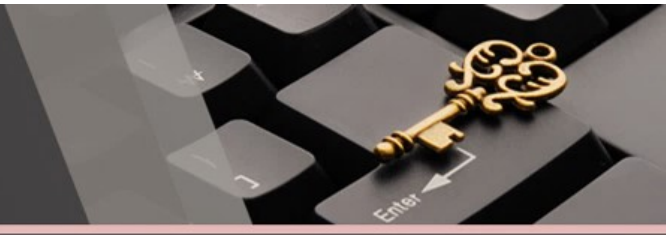
A	B	C	D	E	F	G
0	0	0	0	0	1	0

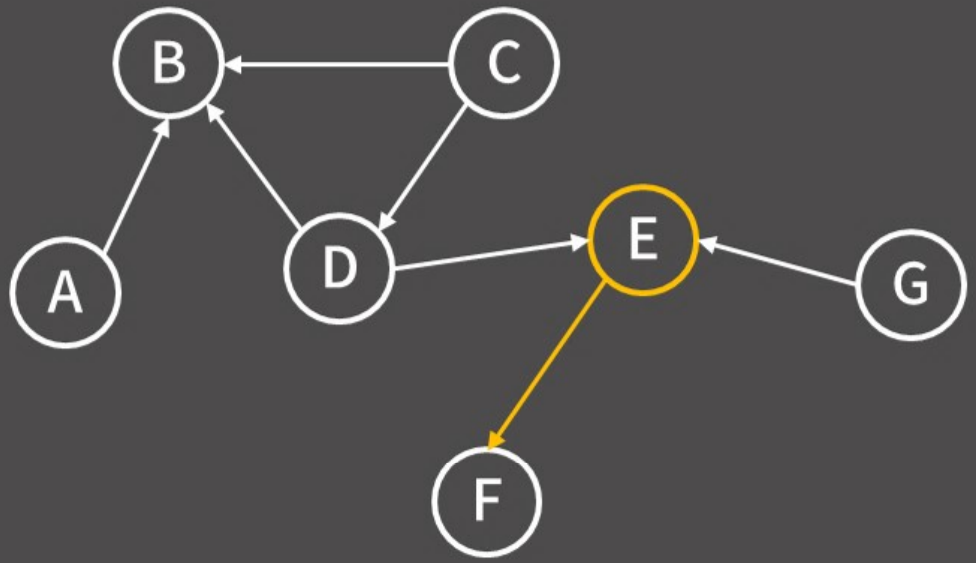
Indegree가 0인 정점을 저장할 큐

E

위상 정렬 결과 A C G D B

0x02 위상 정렬 구현





Indegree 배열

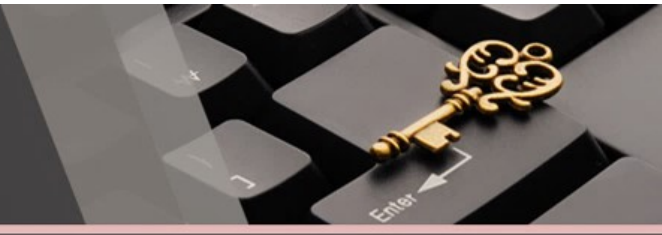
A	B	C	D	E	F	G
0	0	0	0	0	0	0

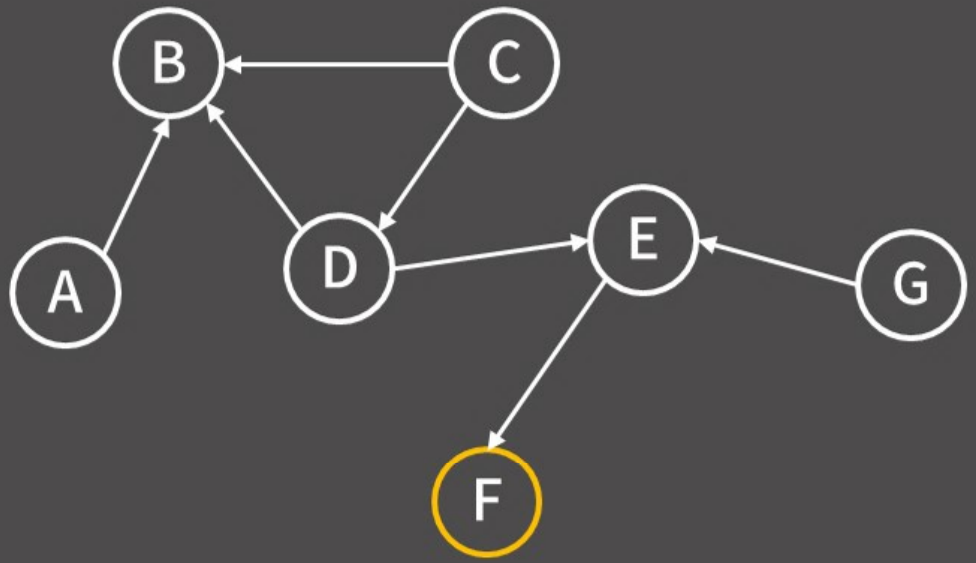
Indegree가 0인 정점을 저장할 큐

F

위상 정렬 결과 A C G D B E

0x02 위상 정렬 구현





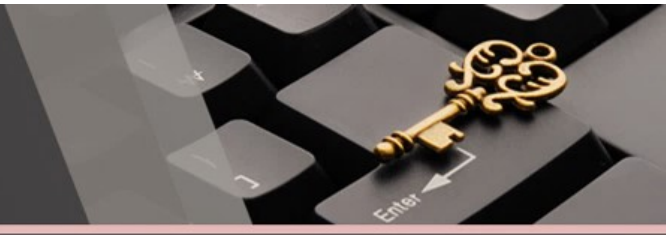
Indegree 배열

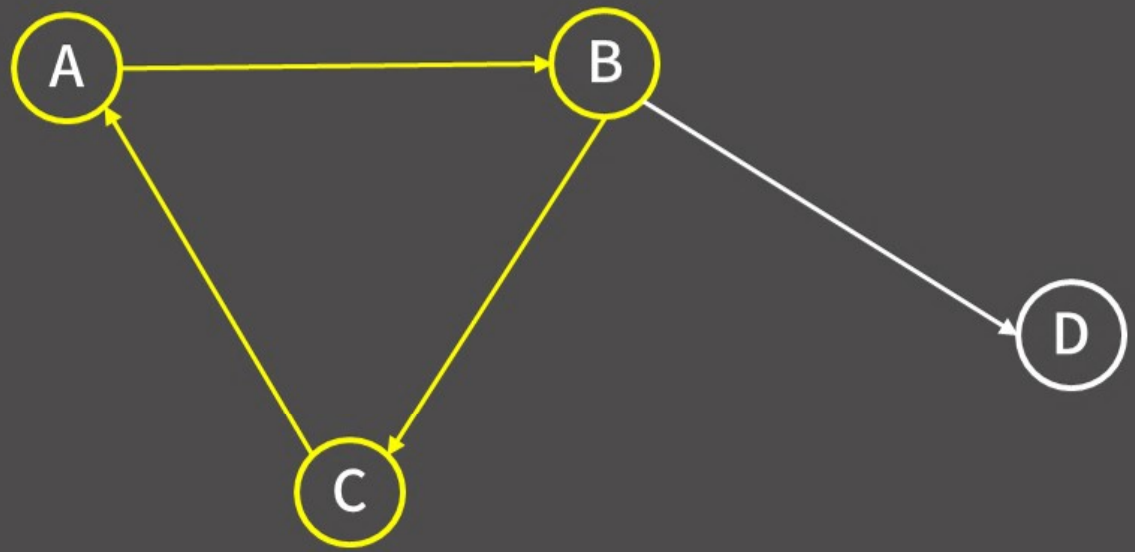
A	B	C	D	E	F	G
0	0	0	0	0	0	0

Indegree가 0인 정점을 저장할 큐

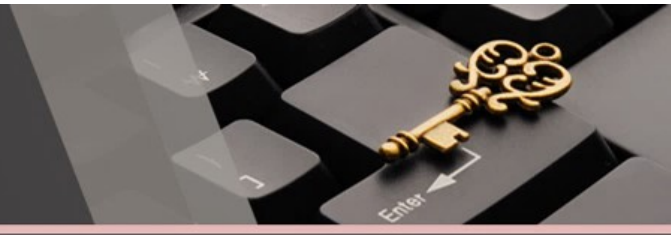
위상 정렬 결과 A C G D B E F

0x02 위상 정렬 구현



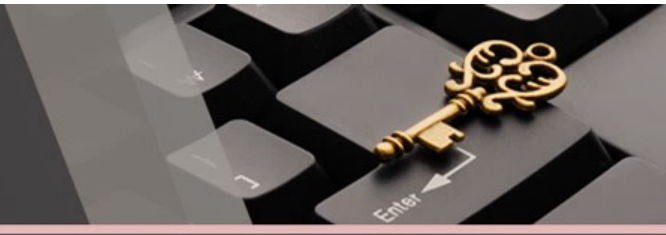


0x02 위상 정렬 구현




```cpp
vector<int> adj[10];
int indeg[10];
int n;
void topo_sort(){
  queue<int> q;
  vector<int> result;
  for(int i = 1; i <= n; i++)
    if(indeg[i] == 0) q.push(i);
  while(!q.empty()){
    int cur = q.front();
    result.push_back(cur);
    for(int i = 0; i < adj[cur].size(); i++){
      int nxt = adj[cur][i];
      indeg[nxt]--;
      if(indeg[nxt] == 0) q.push(nxt);
    }
  }
  if(result.size() != n){
    cout << "cycle exists";
    return;
  }
  for(int i = 0; i < n; i++) cout << result[i] << ' ';
}
```

0x03 연습 문제

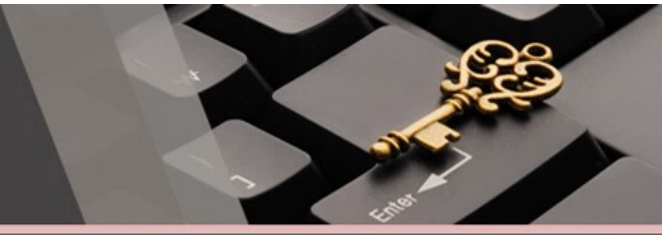


연습 문제 – BOJ 2252번: 줄 세우기

정답 코드 : http://boj.kr/e278ae1c274645d995341fd759a43cba

```
01  cin >> n >> m;
02  while(m--){
03    int a, b;
04    cin >> a >> b;
05    adj[a].push_back(b);
06    indeg[b]++;
07  }
```

강의 정리



- 위상 정렬의 정의를 익혔다.
- 사이클이 없는 방향 그래프(DAG)에서만 올바른 위상 정렬이 존재함을 익혔다.
- 위상 정렬 알고리즘을 구현할 수 있다.