

삼성 청년 SW 아카데미

APS 응용

목차

1. 그래프 탐색 - 너비 우선 탐색(BFS)
2. 너비 우선 탐색(BFS) 구현 - 인접 행렬
3. 너비 우선 탐색(BFS) 구현 - 인접 리스트

그래프 탐색 - 너비 우선 탐색(BFS)

그래프 탐색(순회)

- ✓ 그래프 순회는 비선형구조인 그래프로 표현된 모든 자료(정점)를 빠짐없이 탐색하는 것을 의미한다.
- ✓ 두 가지 방법
 - 너비 우선 탐색(Breadth First Search, BFS)
 - 깊이 우선 탐색(Depth First Search, DFS)

BFS(Breadth First Search)

- ✓ 너비우선탐색은 탐색 시작점의 인접한 정점들을 먼저 모두 차례로 방문한 후에, 방문했던 정점을 시작점으로 하여 다시 인접한 정점들을 차례로 방문하는 방식
- ✓ 인접한 정점들에 대해 탐색을 한 후, 차례로 다시 너비우선탐색을 진행해야 하므로, 선입선출 형태의 자료구조인 큐를 활용함.

✓ 입력 파라미터 : 탐색 시작 정점 v

그래프

```
BFS( $v$ ) // 탐색 시작 정점  $v$ 
    큐 생성
    시작 정점  $v$ 를 큐에 삽입
    정점  $v$ 를 방문한 것으로 표시
    while (큐가 비어 있지 않은 경우) {
         $t \leftarrow$  큐의 첫 번째 원소 반환
        for ( $t$ 와 연결된 모든 간선에 대해){
             $u \leftarrow t$ 의 인접 정점
             $u$ 가 방문되지 않은 곳이면,
             $u$ 를 큐에 넣고, 방문한 것으로 표시
        }
    }
end BFS()
```

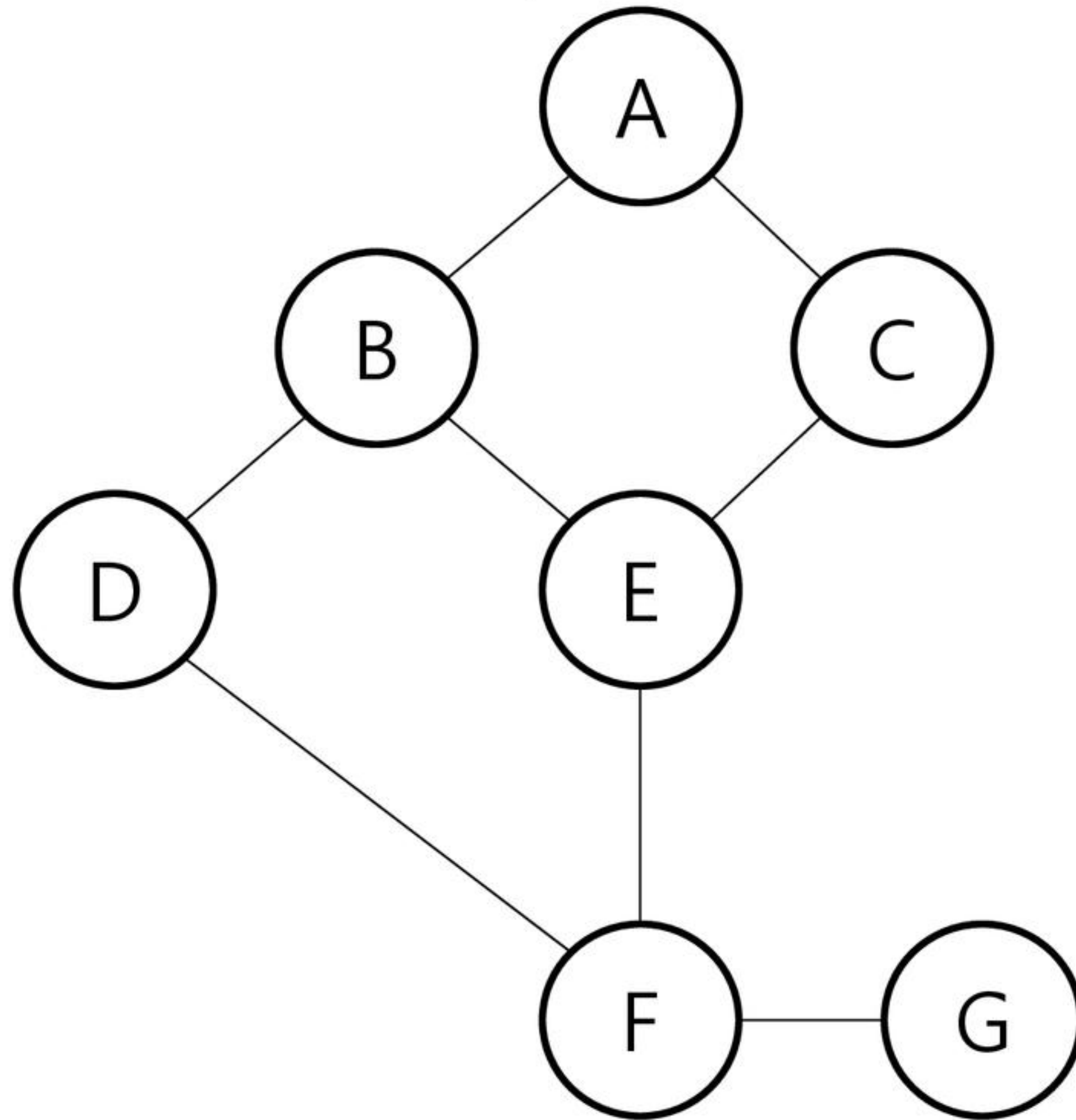
트리

```
BFS()
    큐 생성
    루트  $v$ 를 큐에 삽입
    while (큐가 비어 있지 않은 경우) {
         $t \leftarrow$  큐의 첫 번째 원소 반환

        for ( $t$ 와 연결된 모든 간선에 대해) {
             $u \leftarrow t$ 의 자식노드
             $u$ 를 큐에 삽입
        }
    }
end BFS()
```


✓ 초기 상태

- Visited 배열 생성 및 false로 초기화
- Q 생성
- 시작 정점(A) 방문처리 및 enqueue



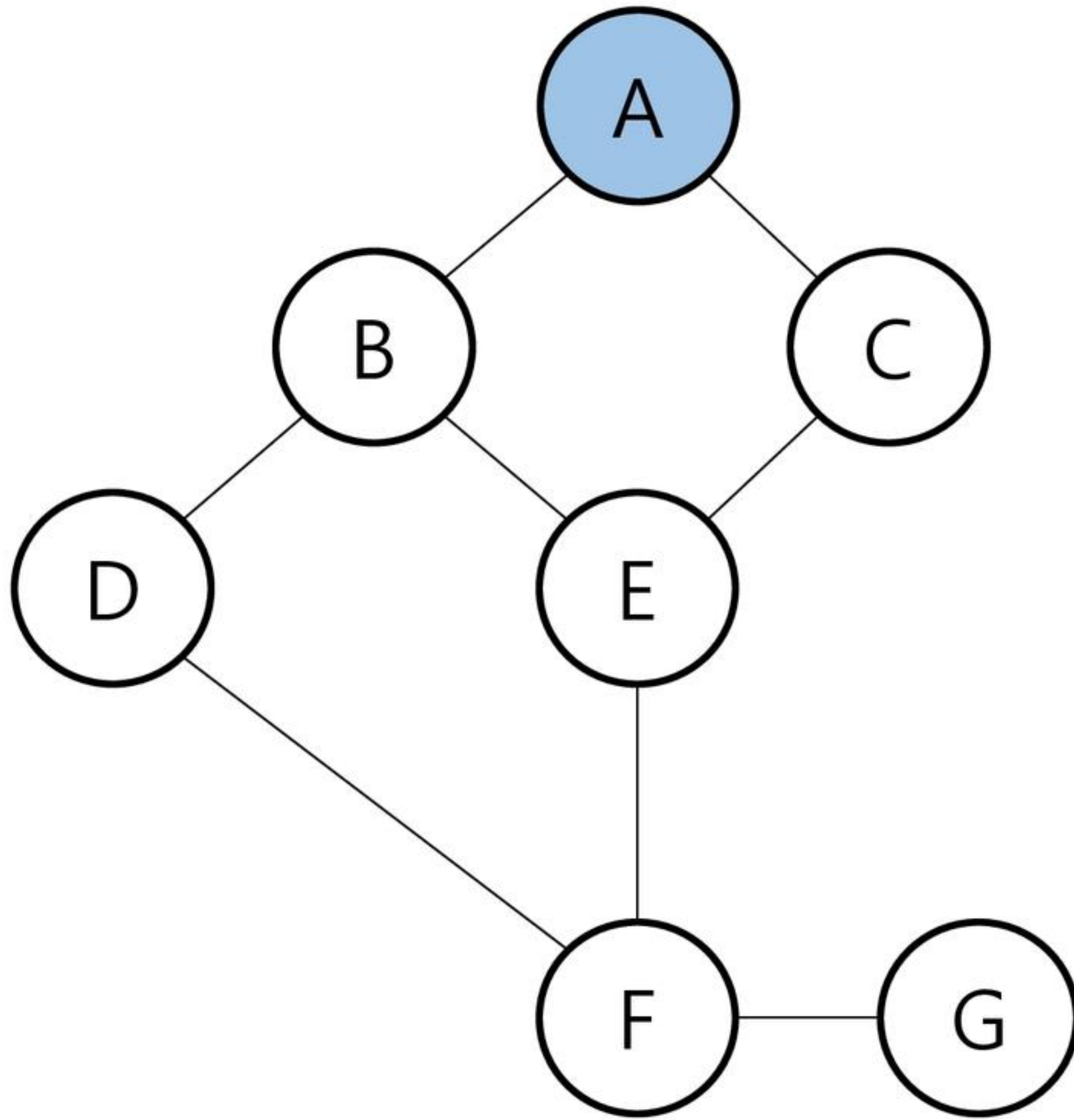
	A	B	C	D	E	F	G
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
Visited	T	F	F	F	F	F	F

Q	A						
---	---	--	--	--	--	--	--

탐색순서							
------	--	--	--	--	--	--	--

✓ 탐색 진행

- dequeue A
- A의 인접 정점 방문처리 및 enqueue



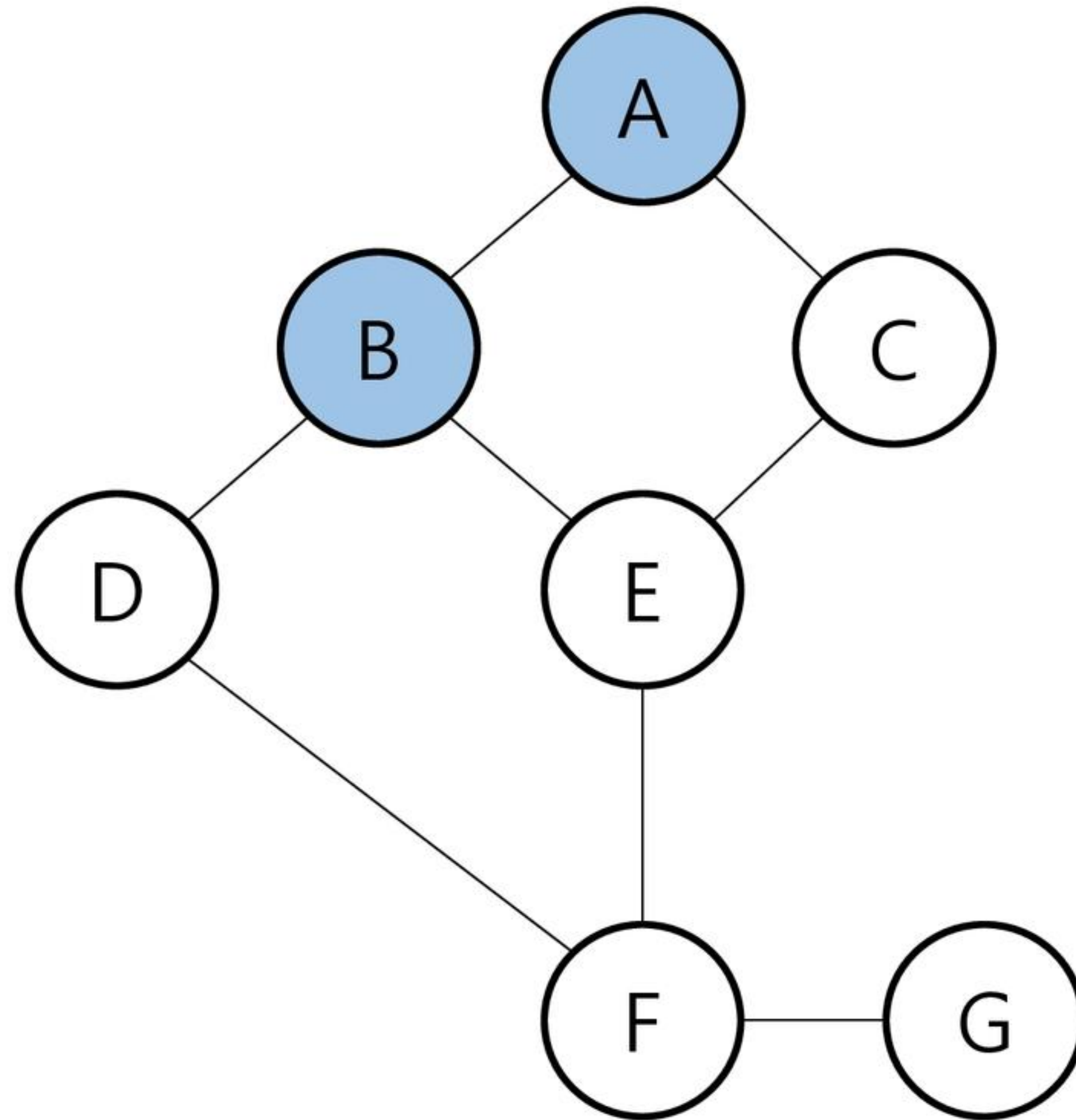
	A	B	C	D	E	F	G
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
Visited	T	T	T	F	F	F	F

Q	B	C					
---	---	---	--	--	--	--	--

탐색순서	A						
------	---	--	--	--	--	--	--

✓ 탐색 진행

- dequeue B
- B의 인접 정점 방문처리 및 enqueue



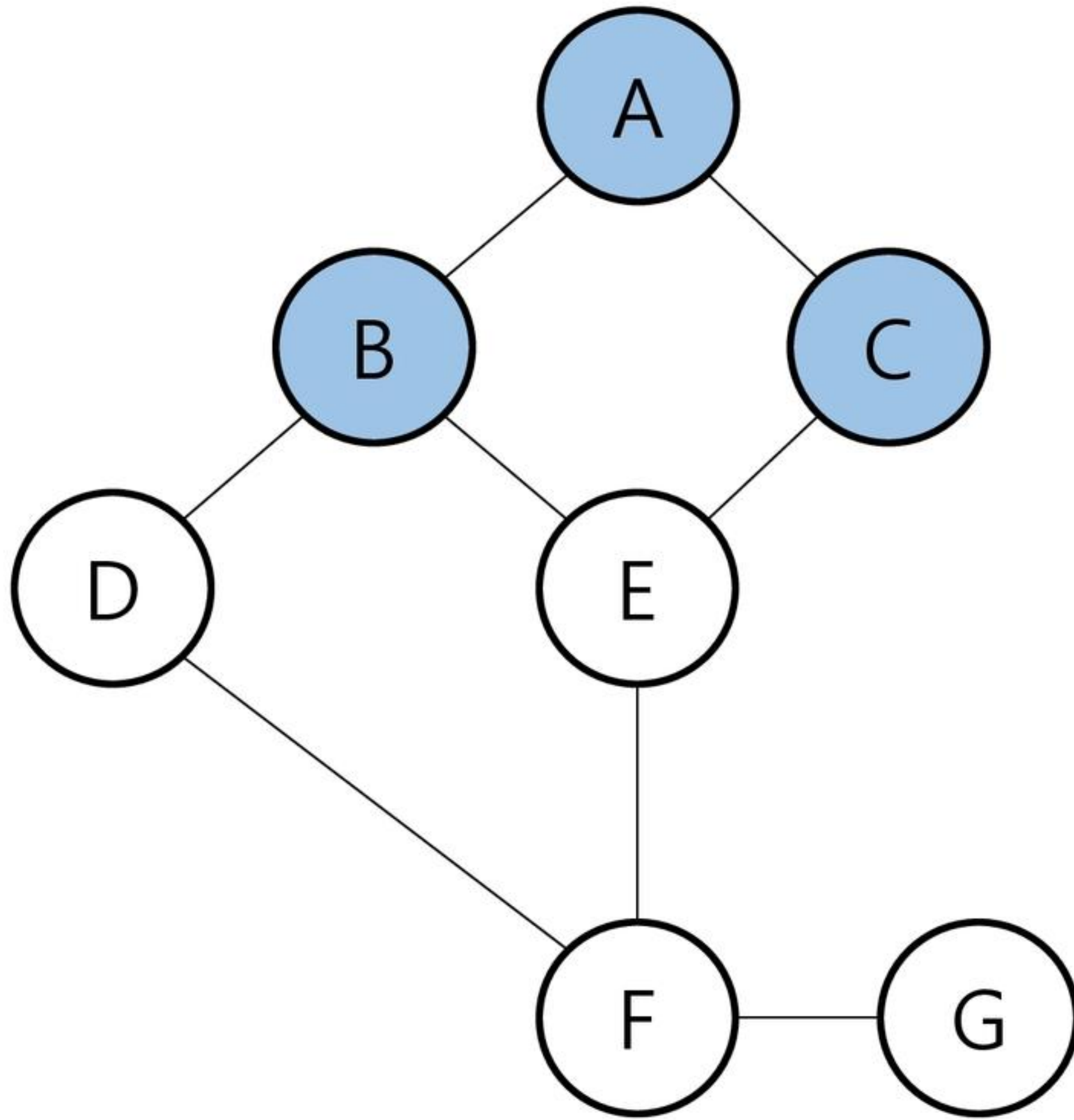
	A	B	C	D	E	F	G
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
Visited	T	T	T	T	T	F	F

Q	C	D	E				
---	---	---	---	--	--	--	--

탐색순서	A	B					
------	---	---	--	--	--	--	--

✓ 탐색 진행

- dequeue C
- C의 인접 정점 방문처리 및 enqueue : 없음



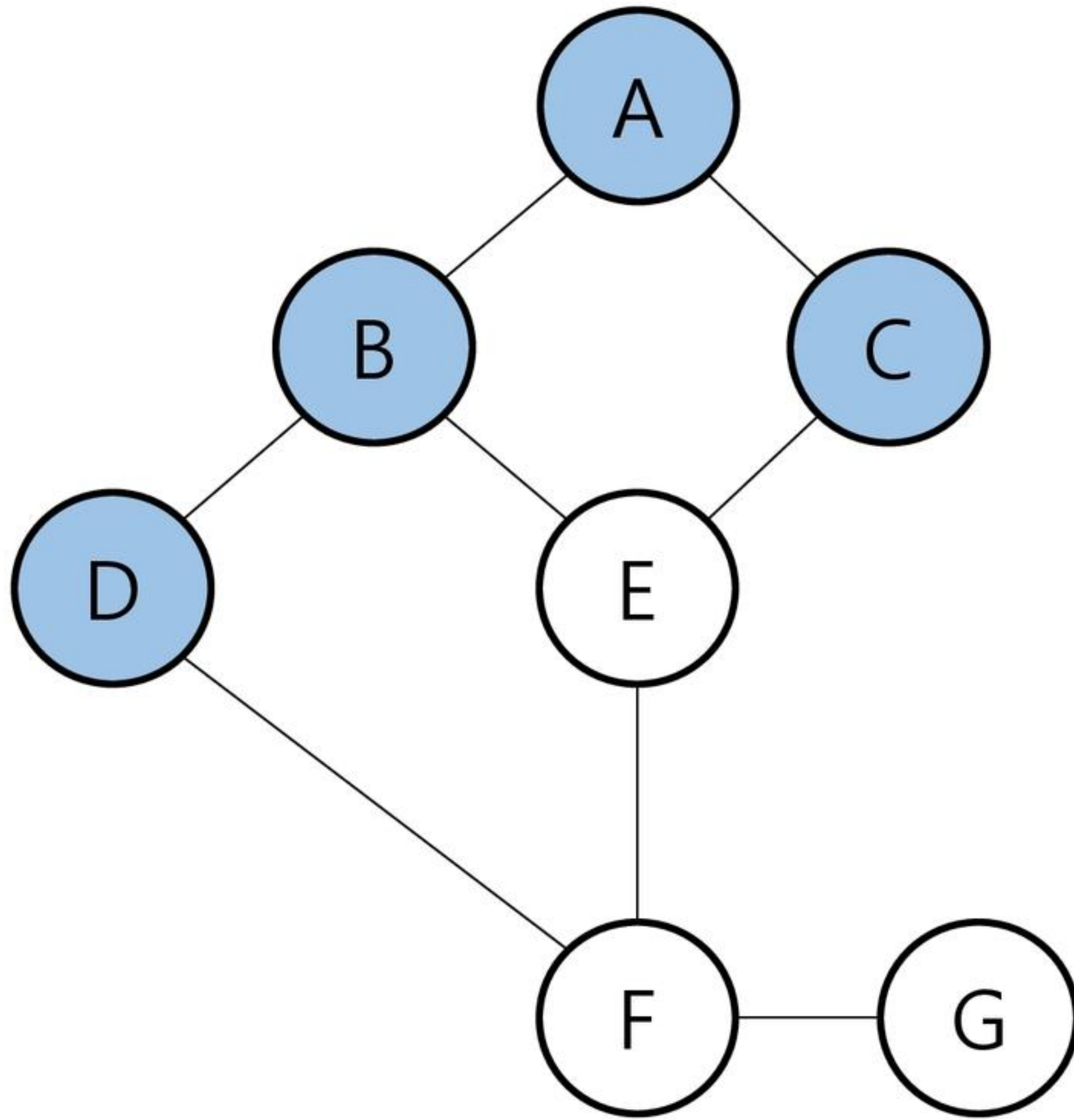
	A	B	C	D	E	F	G
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
Visited	T	T	T	T	T	F	F

Q	D	E					
---	---	---	--	--	--	--	--

탐색순서	A	B	C				
------	---	---	---	--	--	--	--

✓ 탐색 진행

- dequeue D
- D의 인접 정점 방문처리 및 enqueue



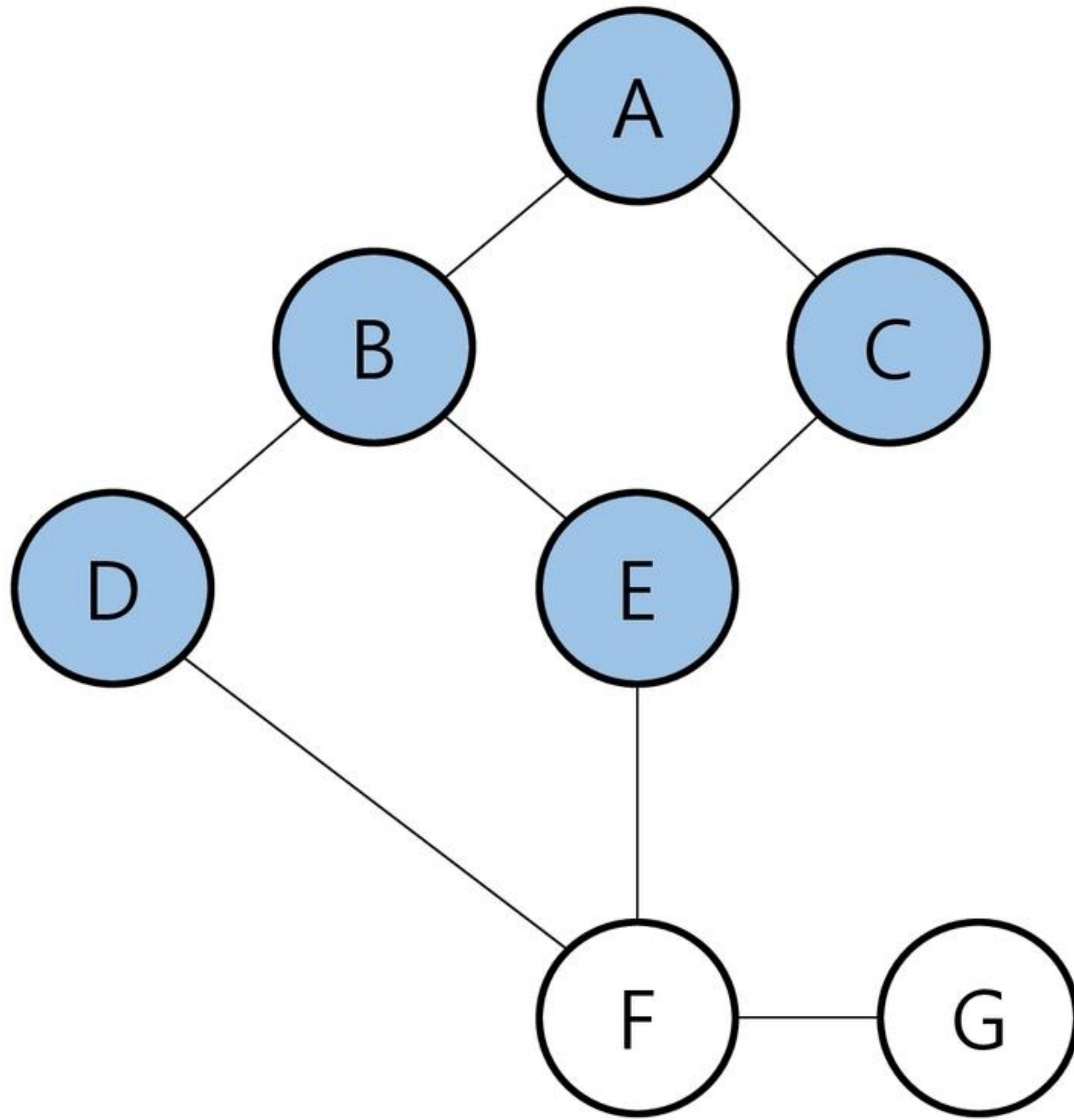
	A	B	C	D	E	F	G
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
Visited	T	T	T	T	T	T	F

Q	E	F					
---	---	---	--	--	--	--	--

탐색순서	A	B	C	D			
------	---	---	---	---	--	--	--

✓ 탐색 진행

- dequeue E
- E의 인접 정점 방문처리 및 enqueue : 없음



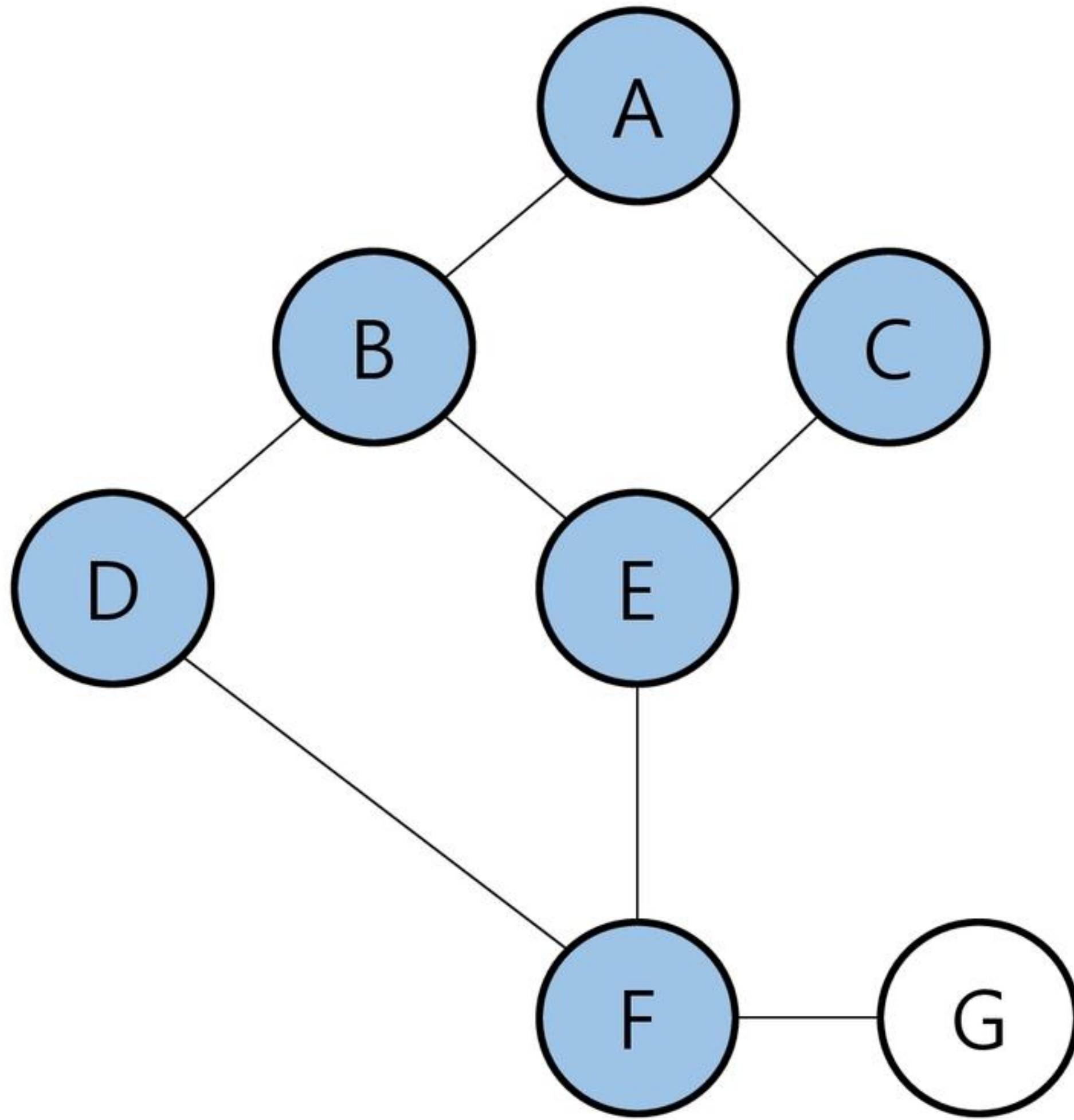
	A	B	C	D	E	F	G
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
Visited	T	T	T	T	T	T	F

Q	F						
---	---	--	--	--	--	--	--

탐색순서	A	B	C	D	E		
------	---	---	---	---	---	--	--

✓ 탐색 진행

- dequeue F
- F의 인접 정점 방문처리 및 enqueue



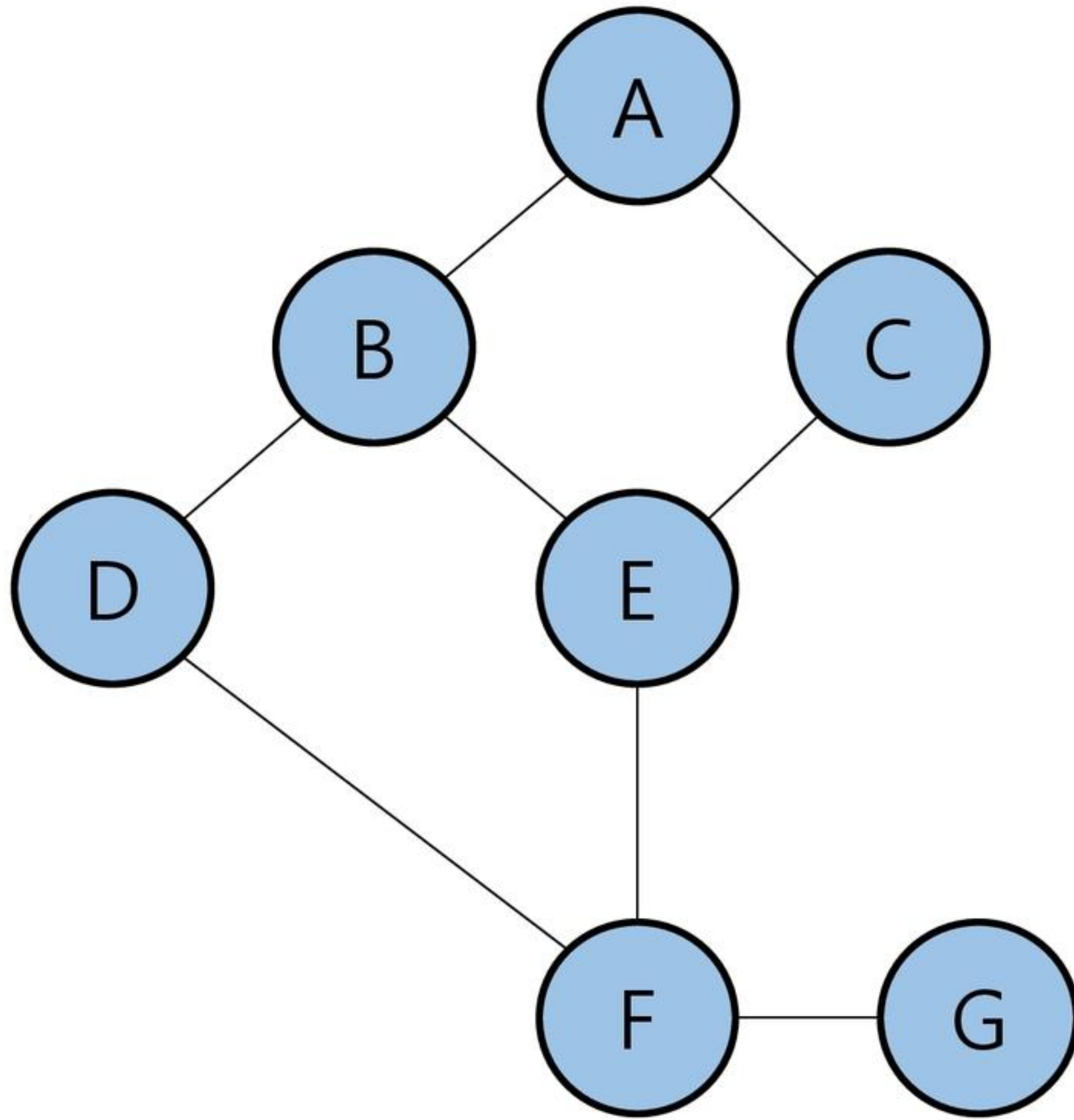
	A	B	C	D	E	F	G
	[0]	[1]	[2]	[3]	[4]	[5]	[6]
Visited	T	T	T	T	T	T	T

Q	G						
---	---	--	--	--	--	--	--

탐색순서	A	B	C	D	E	F	
------	---	---	---	---	---	---	--

✓ 탐색 진행

- dequeue G
- G의 인접 정점 방문처리 및 enqueue : 없음



✓ Q가 비었으므로 탐색 종료

A	B	C	D	E	F	G
[0]	[1]	[2]	[3]	[4]	[5]	[6]

Visited

T	T	T	T	T	T	T
---	---	---	---	---	---	---

Q

--	--	--	--	--	--	--

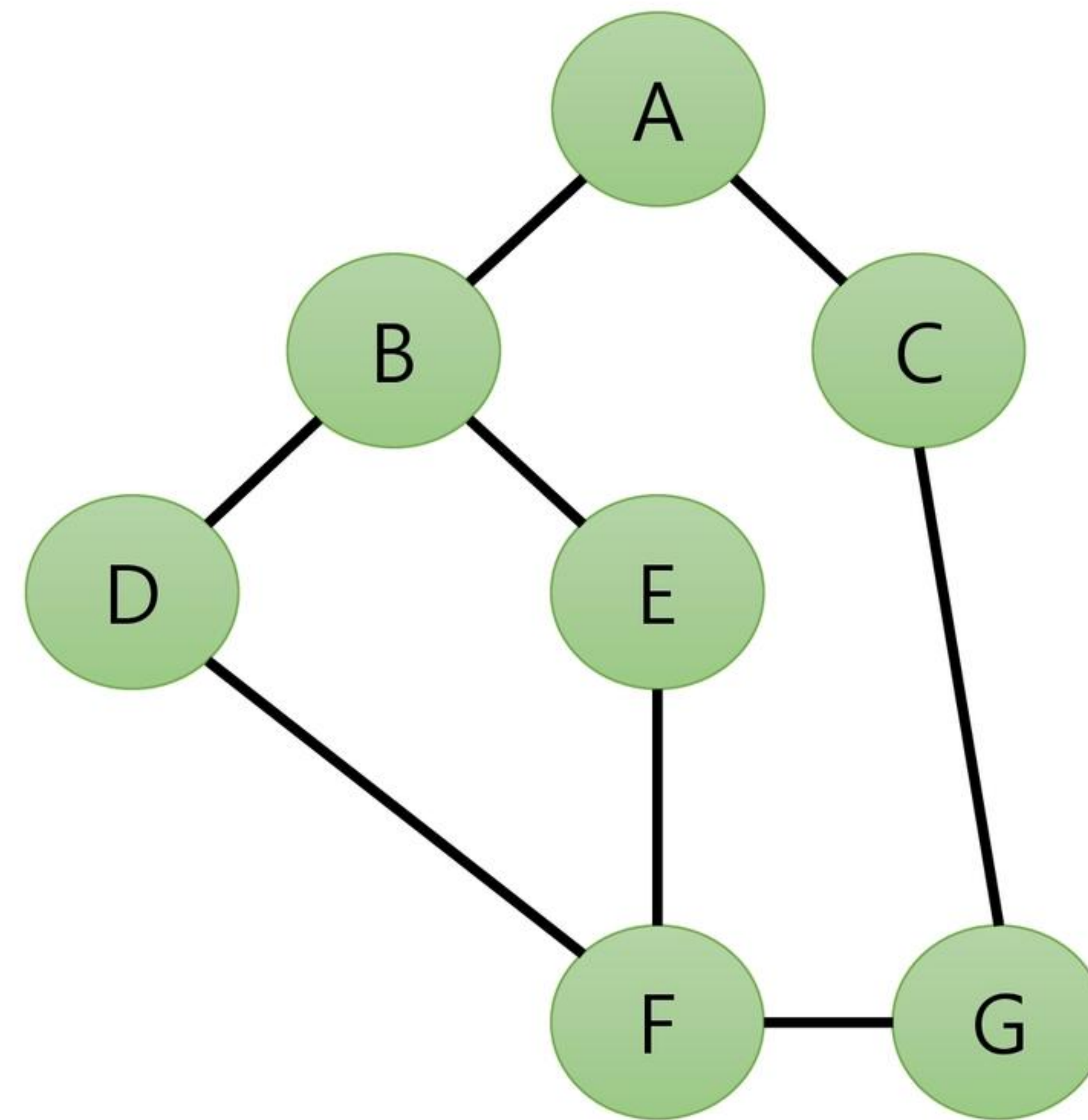
탐색순서

A	B	C	D	E	F	G
---	---	---	---	---	---	---

연습문제 - BFS

- 다음은 연결되어 있는 두 개의 정점 사이의 간선을 순서대로 나열 해놓은 것이다. 모든 정점을 너비 우선 탐색하여 너비 우선 탐색 경로를 출력하시오.
- 단, 시작 정점을 'A'로 시작하시오.

7
8
0 1
0 2
1 3
1 4
3 5
4 5
5 6
2 6



탐색경로 : A B C D E G F

너비 우선 탐색(BFS) 구현 - 인접 행렬

너비 우선 탐색(BFS) 구현 - 인접 행렬

```
// 정점수: N, 인접행렬: adjMatrix, 탐색 시작 정점: start
Queue<Integer> queue = new ArrayDeque<Integer>();
boolean visited[] = new boolean[N];

queue.offer(start);
visited[start] = true;

while(!queue.isEmpty()) {
    int current = queue.poll();
    for(int i=0; i<N; ++i) {
        if(adjMatrix[current][i] && !visited[i]) {
            queue.offer(i);
            visited[i] = true;
        }
    }
}
```


너비 우선 탐색(BFS) 구현 - 인접 리스트

너비 우선 탐색(BFS) 구현 - 인접 리스트

```
// 정점수: N, 인접리스트: adjList, 탐색 시작 정점: start
Queue<Integer> queue = new ArrayDeque<Integer>();
boolean visited[] = new boolean[N];

queue.offer(start);
visited[start] = true;

while(!queue.isEmpty()) {
    int current = queue.poll();
    for(Node temp = adjList[current]; temp != null; temp = temp.next) {
        if(!visited[temp.vertex]) {
            queue.offer(temp.vertex);
            visited[temp.vertex] = true;
        }
    }
}
```


다음 방송에서 만나요!

삼성 청년 SW 아카데미