

2019 UOSPC 문제 해설



Hojun Lee
Dept. of Computer Science & Statistics
University of Seoul

Contents

- Div.2 : 메자이 풀스택
- Div.2 : 근손실 싫어! 단백질 좋아!
- Div.1 : 해외 여행

메자이 풀스택

AC Rate : 43.1%



4 / 1 / 2



7 / 3 / 2

K K K K D D 를 어떻게 배열할 것인가?

→ D를 먼저 배열하자!

→ K와 A의 가중치의 합이 25를 넘는가?

```
if 4*(k2-k1) + 2*(a2-a1) >= 25:
    print("YES")
else:
    print("NO")
```

근손실 싫어! 단백질 좋아! 💪

AC Rate : 51.4%

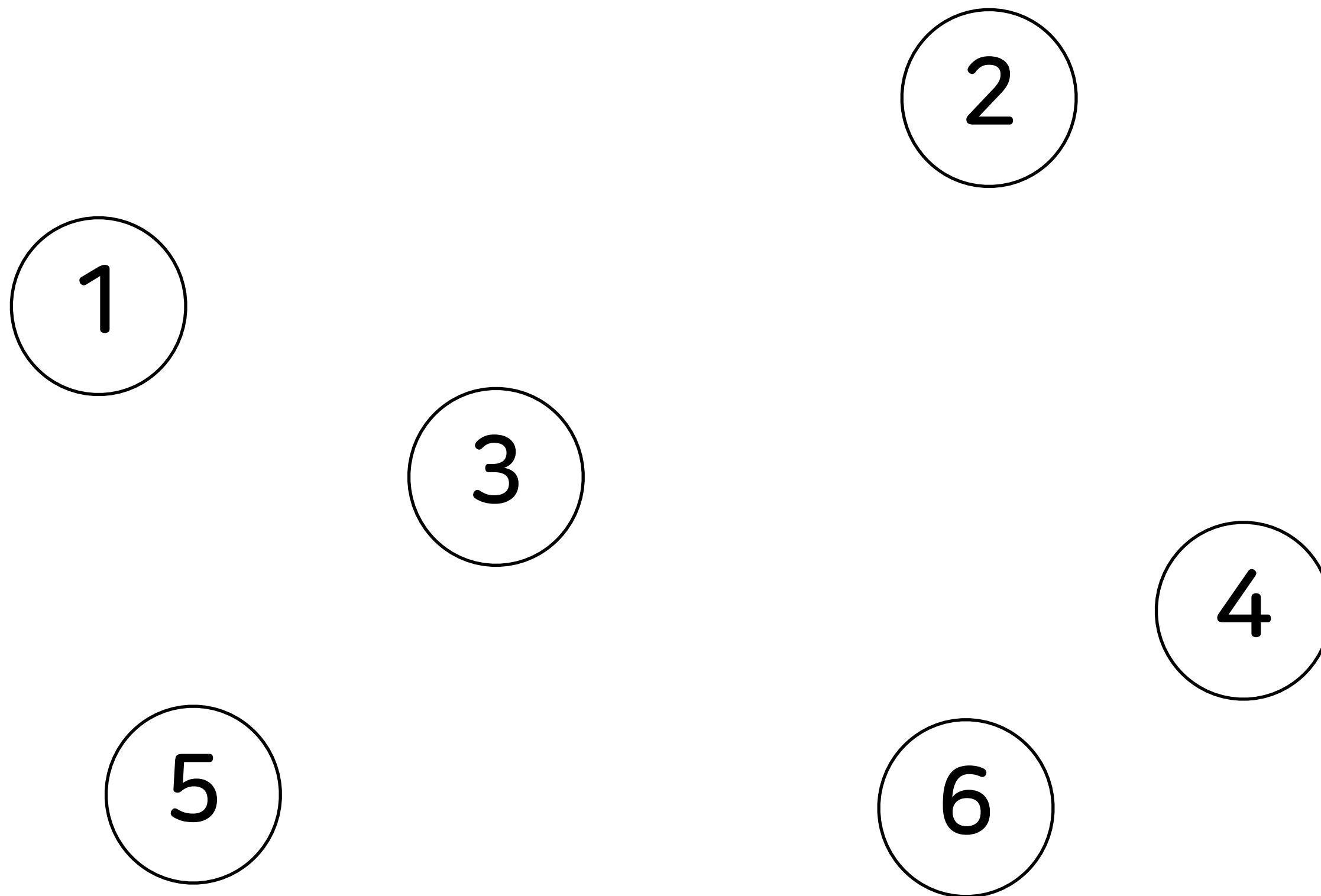
정보량이 많은 문제 → 생각보다 어렵지 않음

주어진 문자열을 DNA → mRNA 전사 규칙에 의해 변환

개시코돈에 해당하는 부분문자열과 종결코돈에 해당하는 부분문자열 찾기!

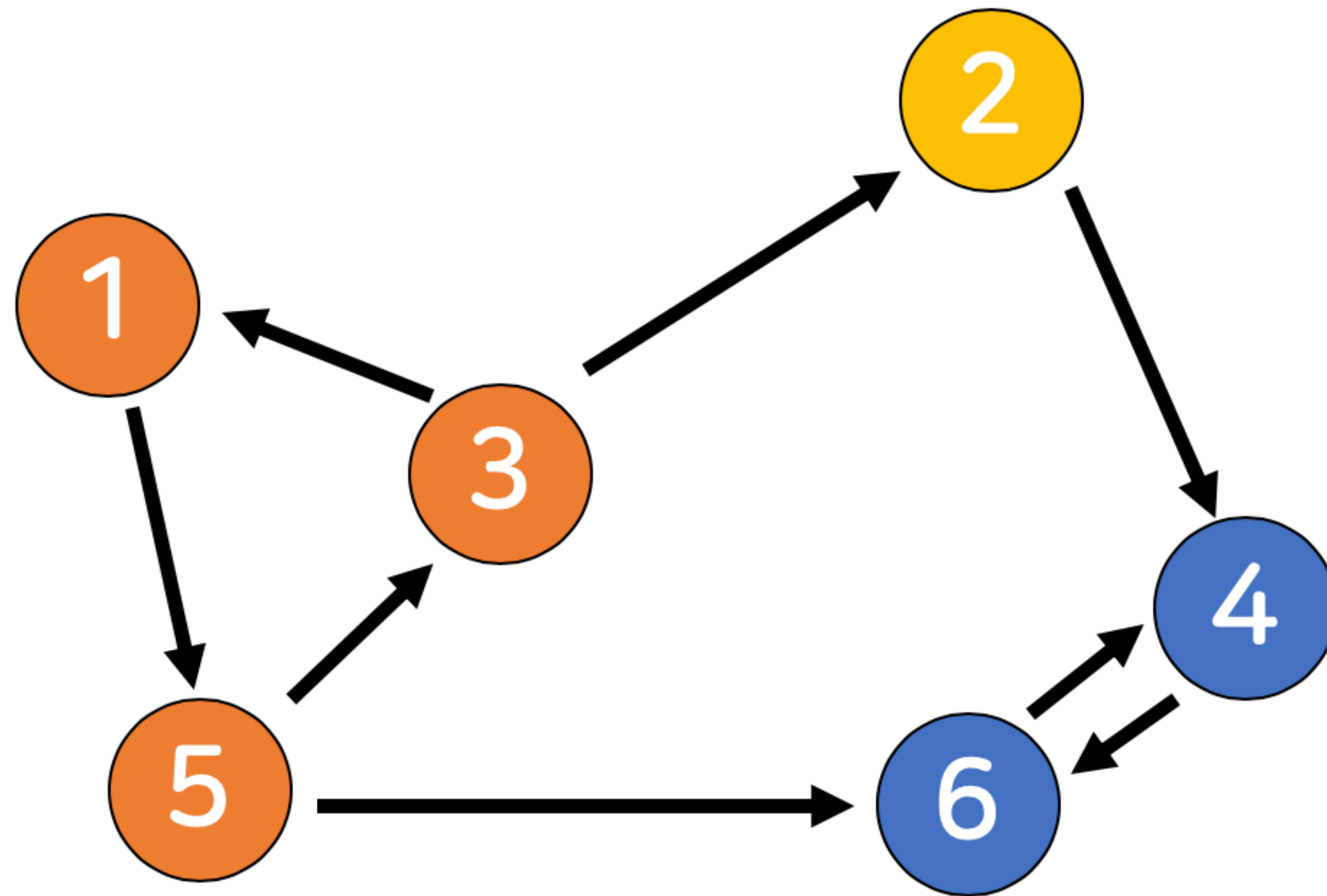
해외 여행 ✈️

AC Rate : 4.9%



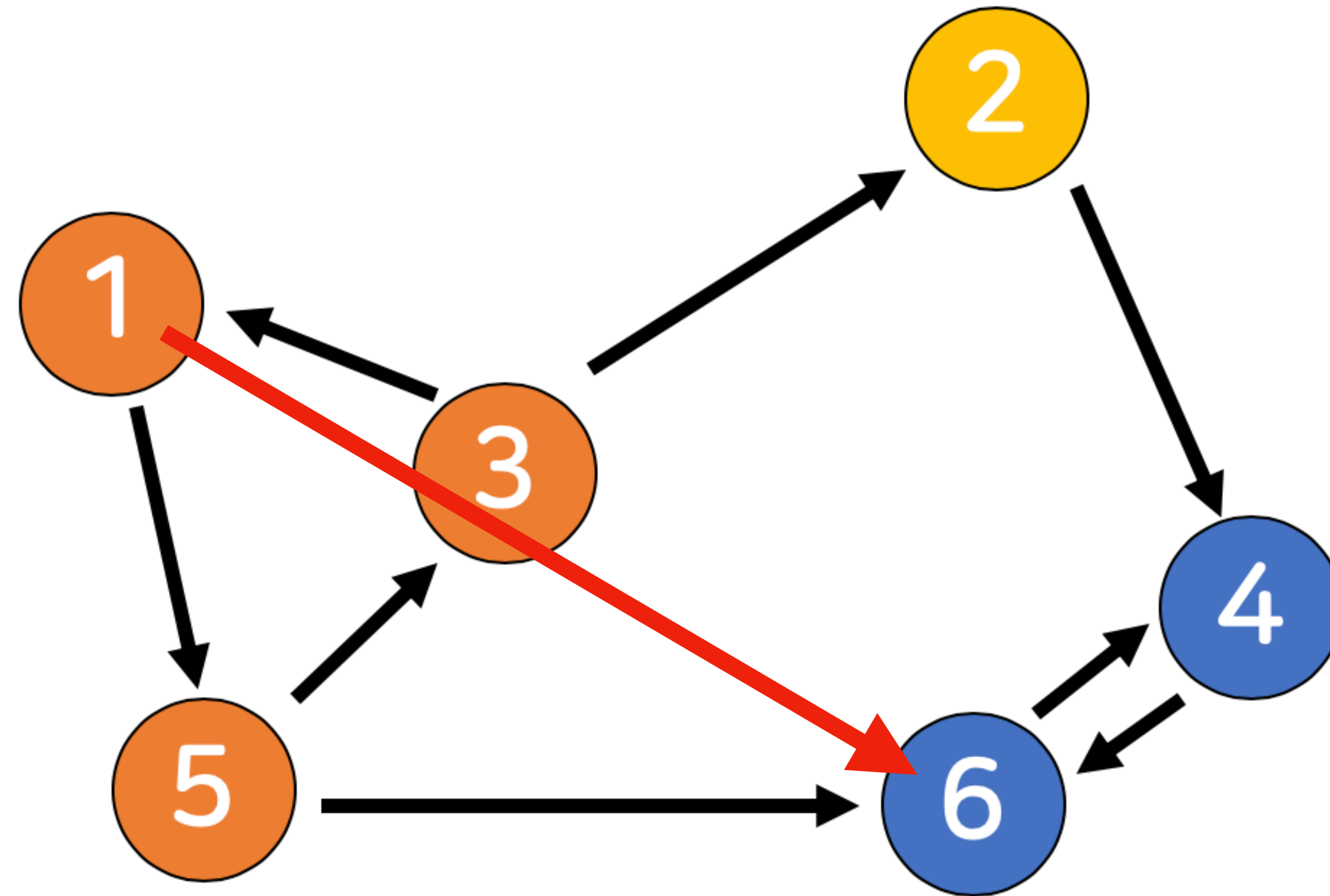
해외 여행 ✈️

AC Rate : 4.9%



해외 여행 ✈️

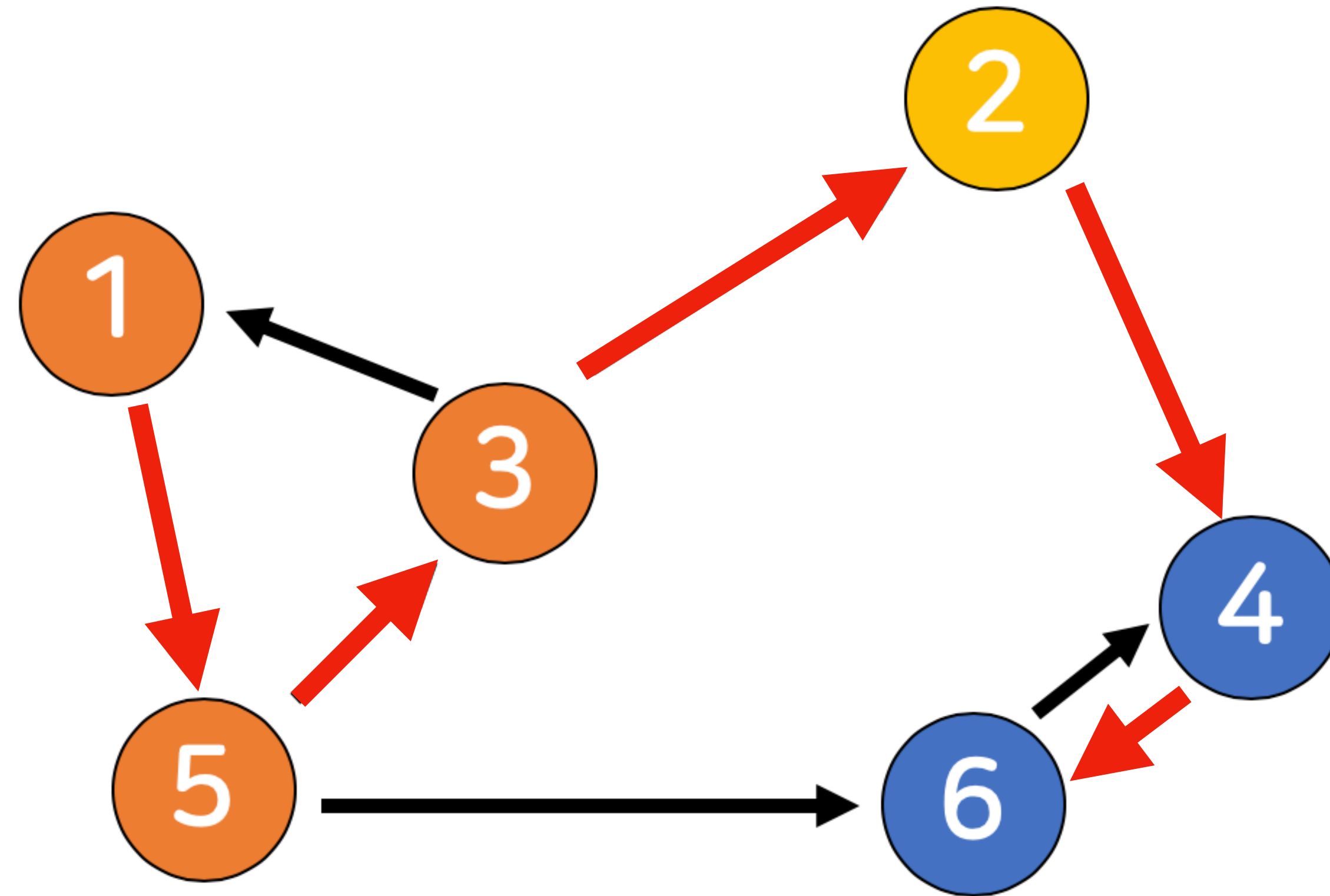
AC Rate : 4.9%



어떻게 도시 1에서 도시 N_c 로 이동할 수 있을까?

해외 여행 ✈️

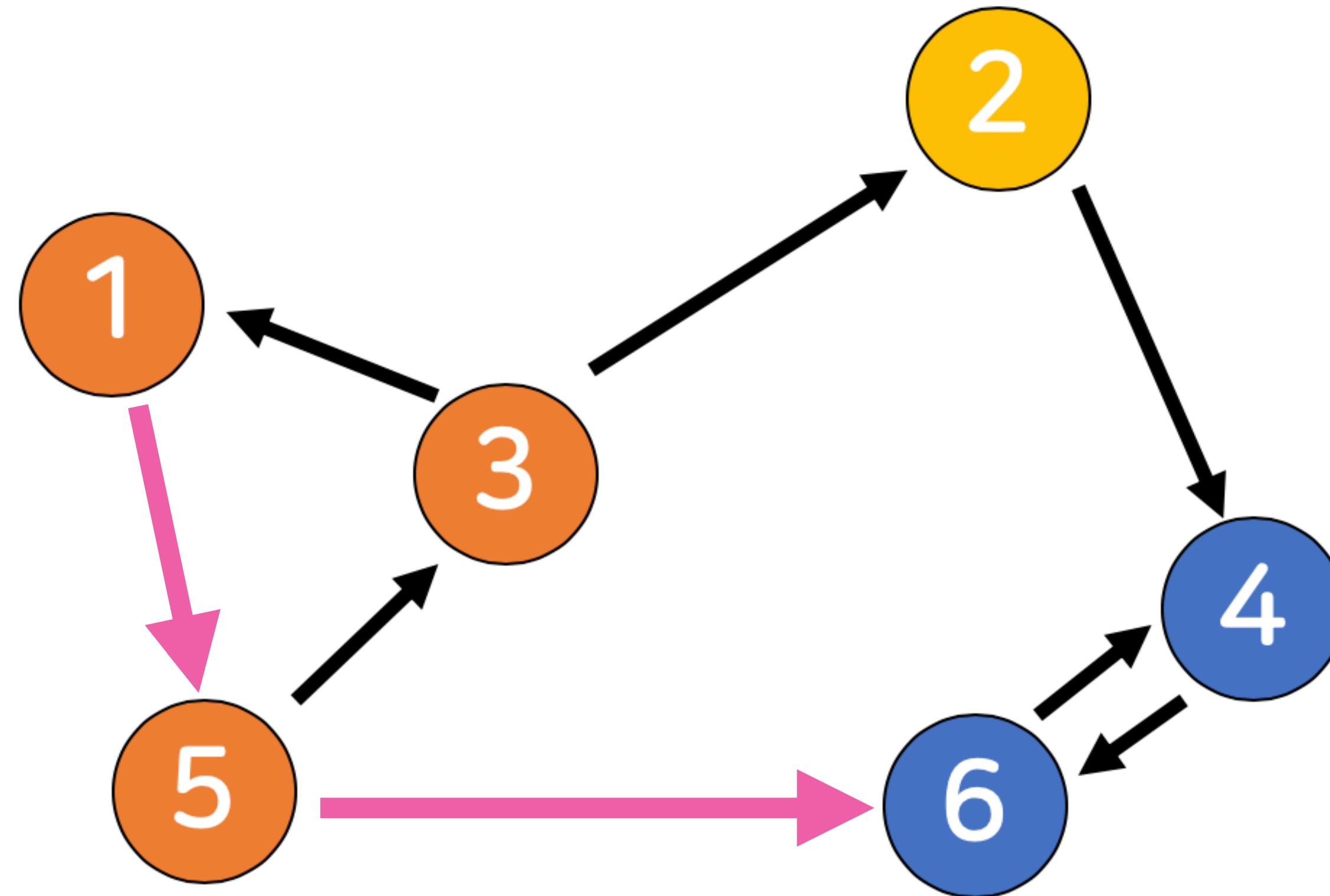
AC Rate : 4.9%



Route 1 : 주황 - 노랑 - 파랑

해외 여행 ✈️

AC Rate : 4.9%



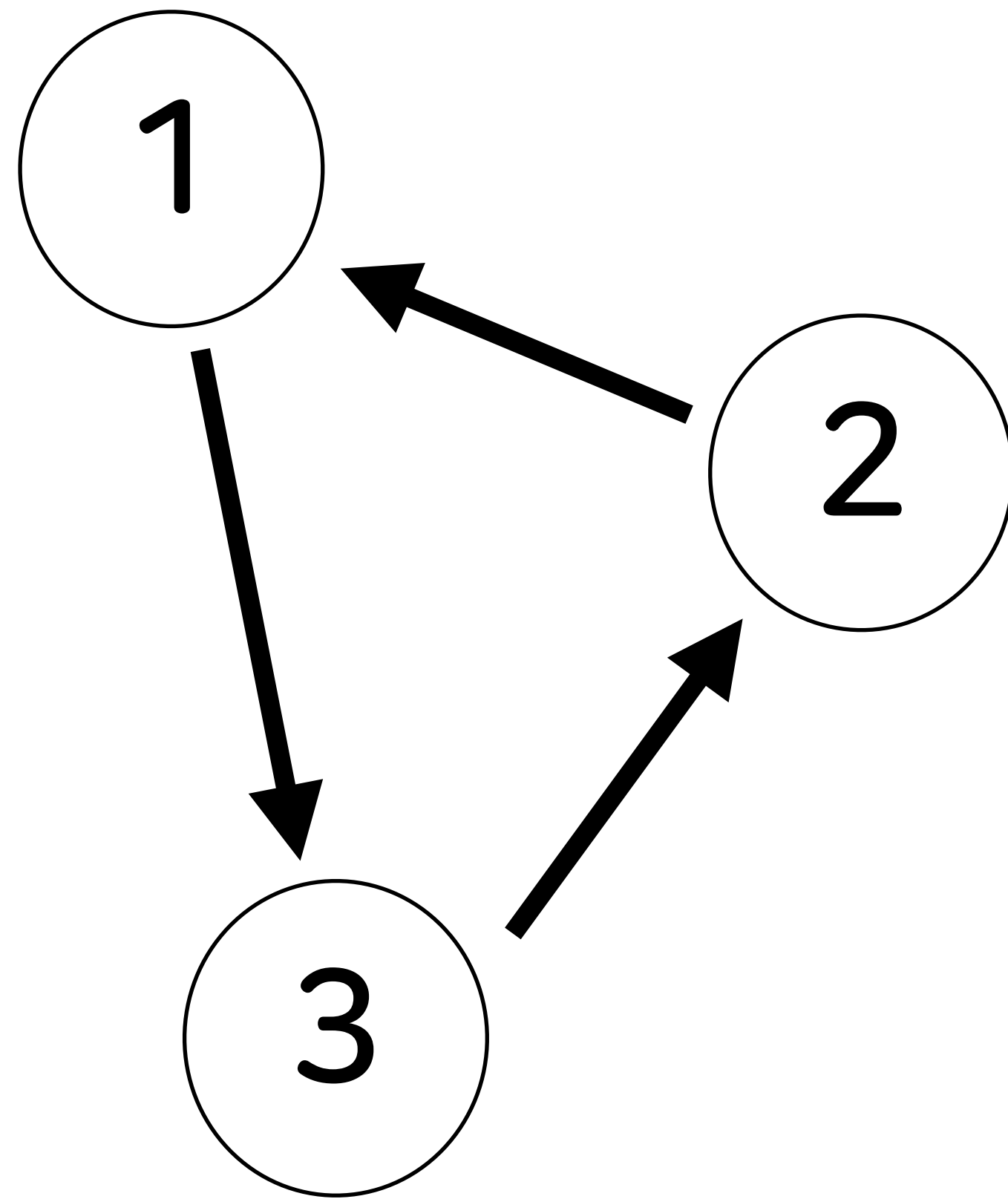
Route 2 : 주황 - 파랑

더 적은 국가를 거치므로 좋은 Route! → Answer : 1

이 문제의 2가지 쟁점

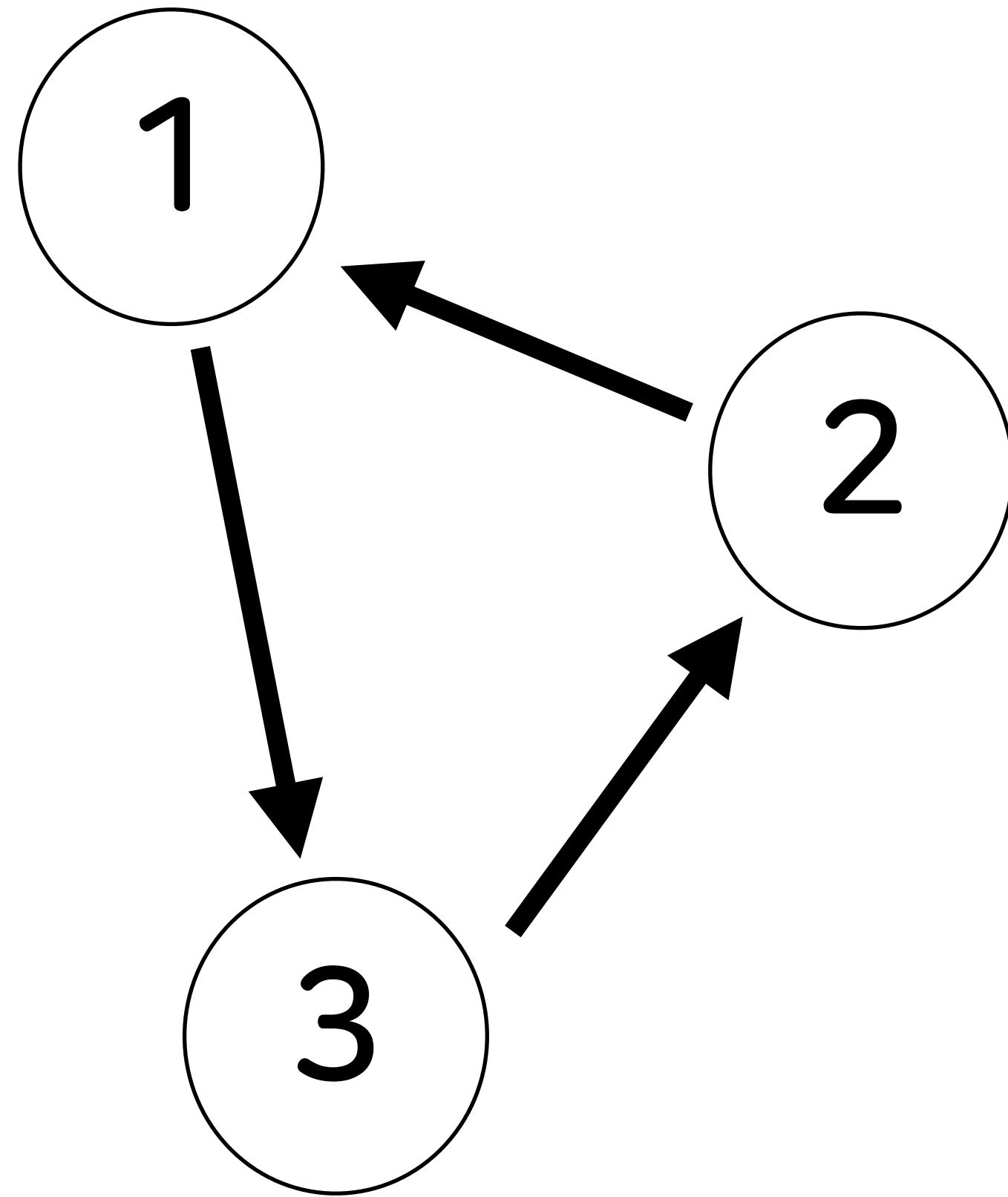
1. 두 도시 C_i, C_j 가 같은 국가에 속했는지 어떻게 알까?
2. 두 도시가 이동 가능한지는 어떻게 알 수 있을까?

국가의 조건



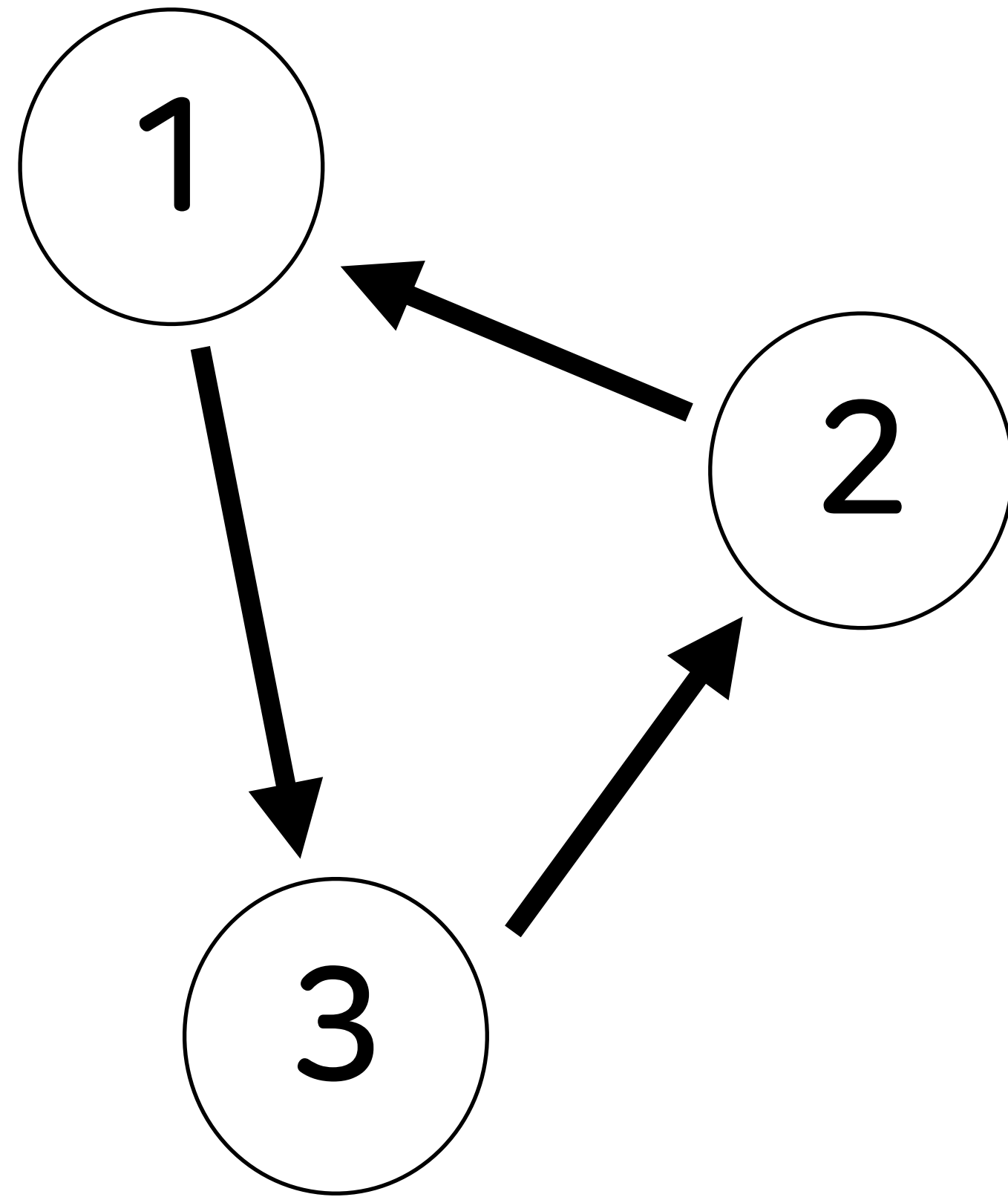
같은 국가에 속한 국가는 서로 도달 가능

국가의 조건



같은 국가에 속한 국가는 서로 도달 가능
서로 도달 가능하지 않으면 다른 국가!

SCC(Strongly Connected Component)

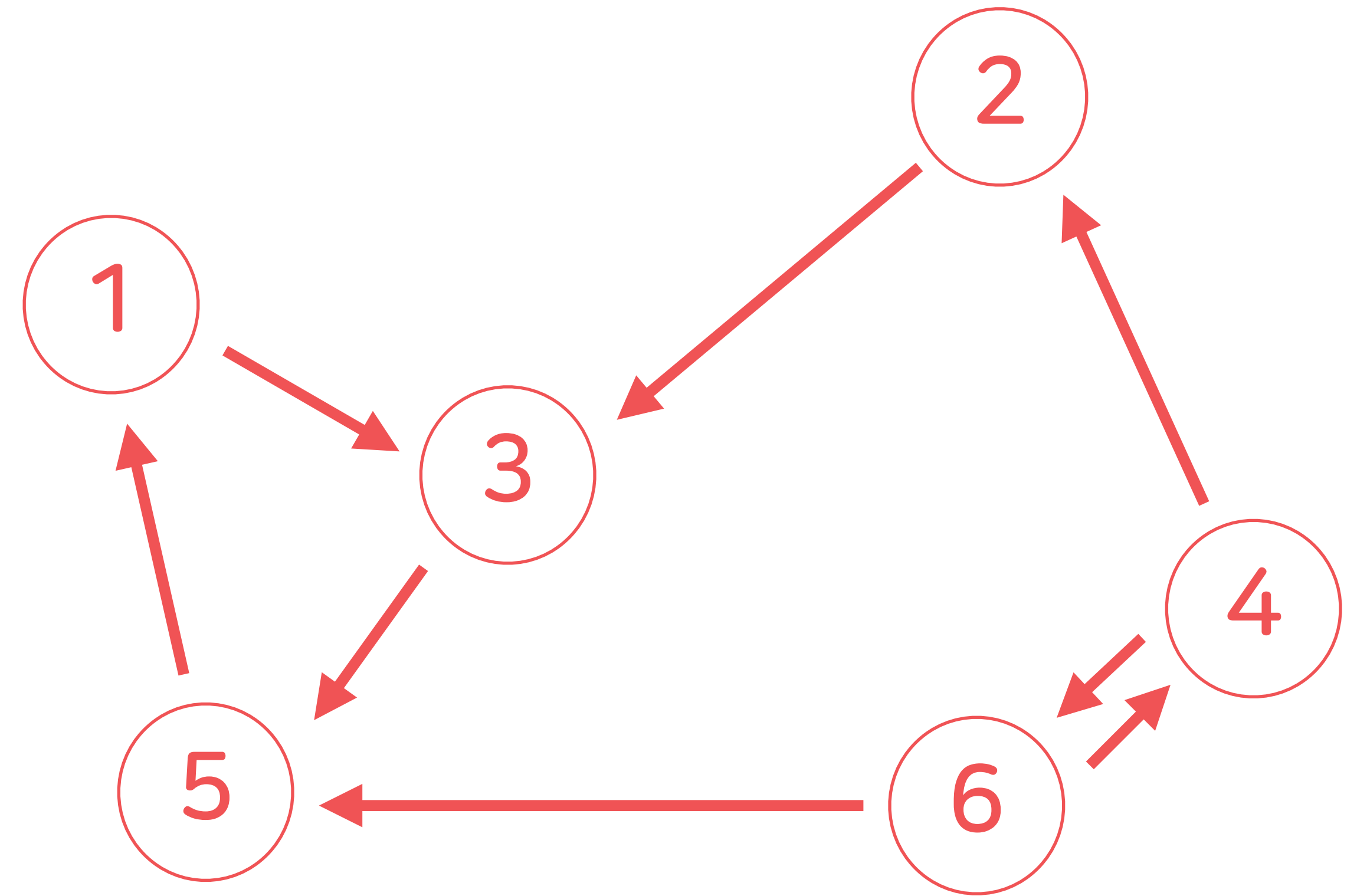
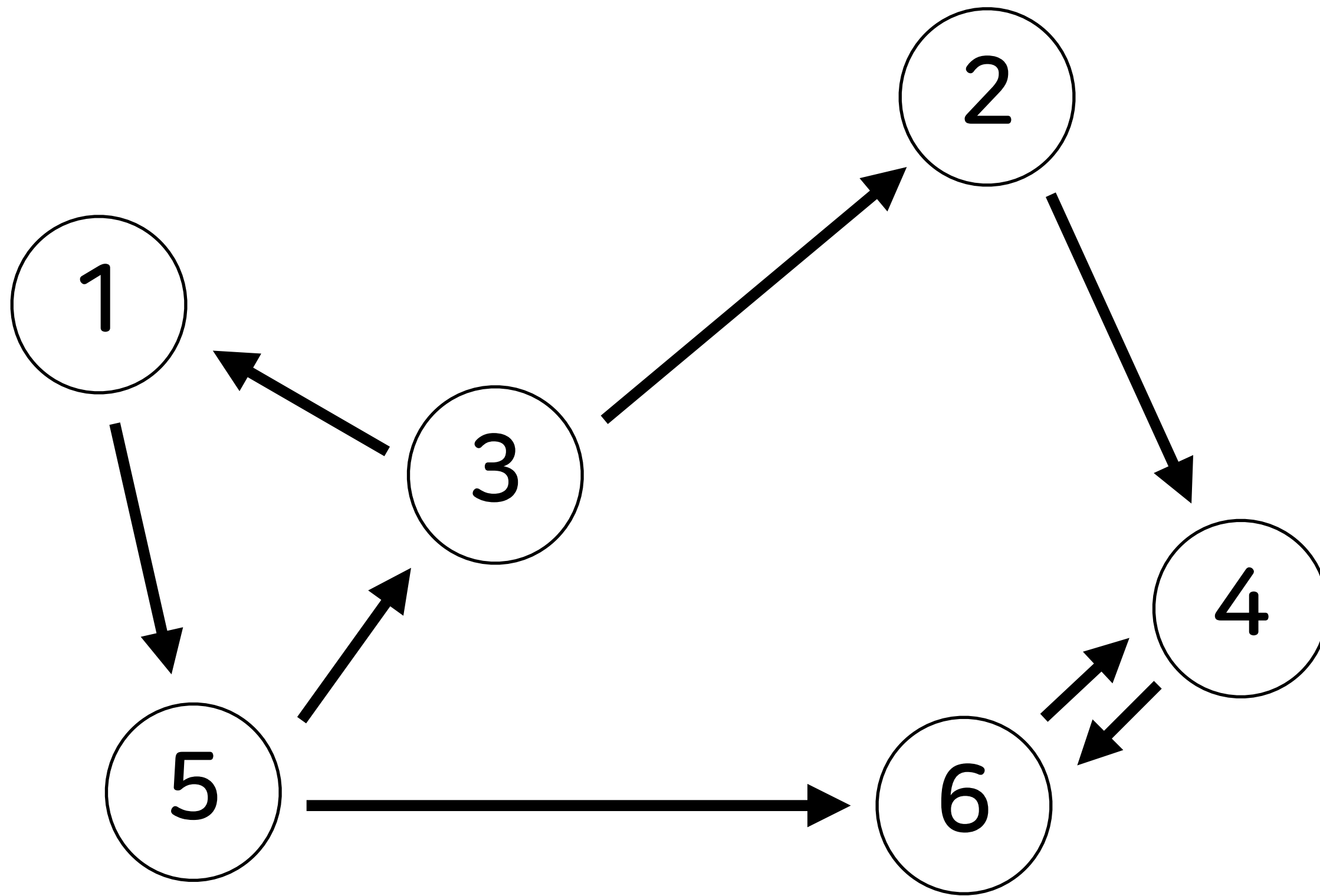


방향 그래프에서 서로 도달 가능한 Node의 집합

$\{C_1, C_2, \dots, C_n\}$

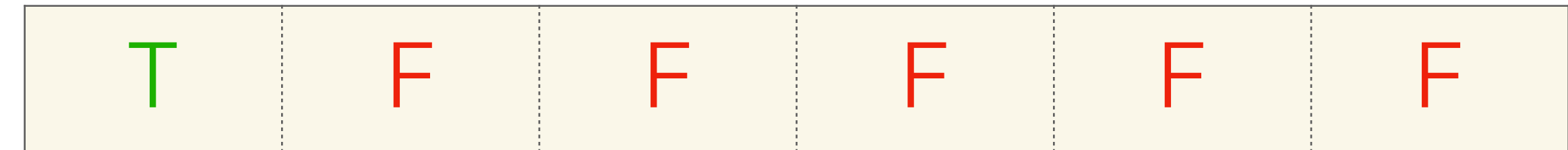
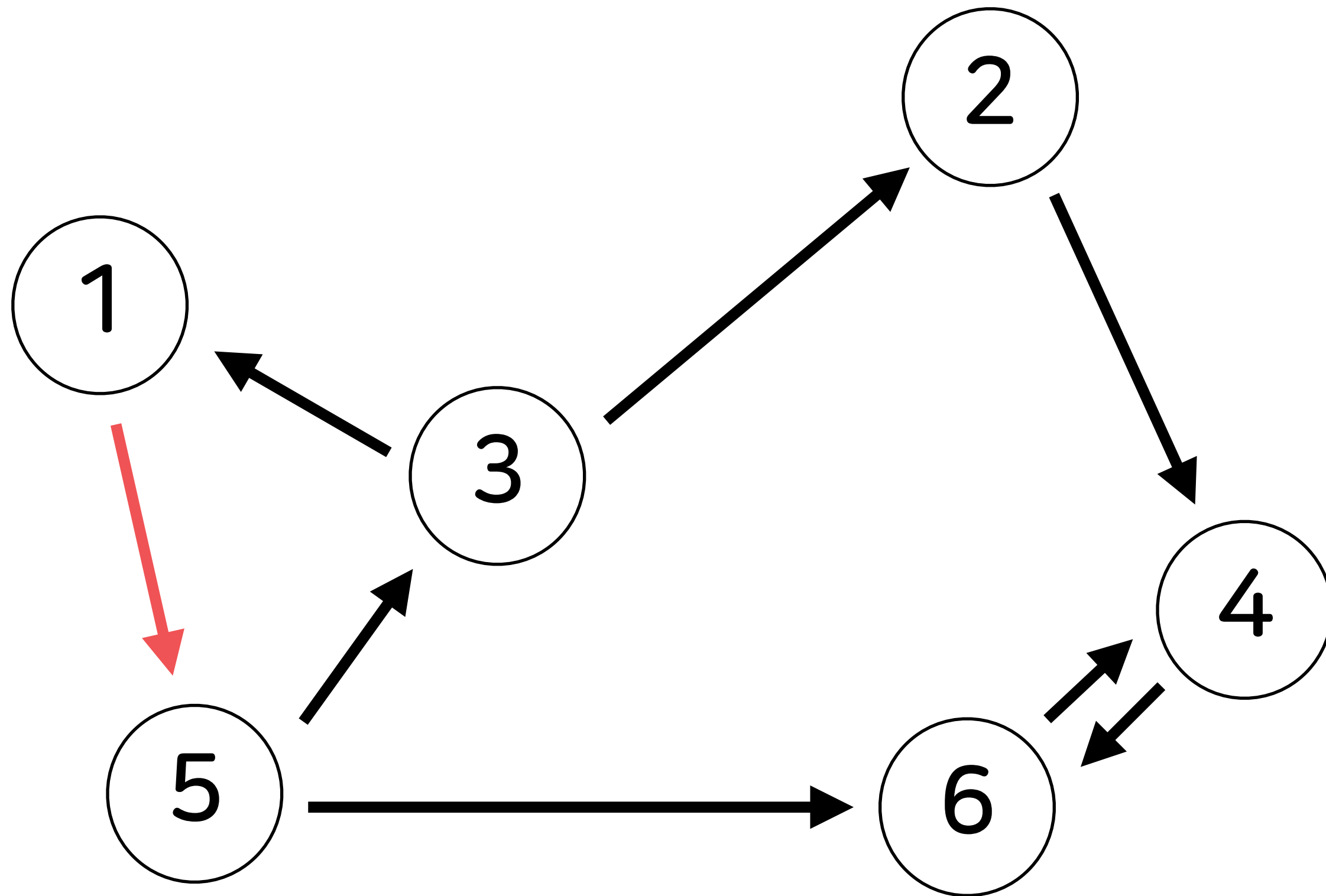
을 SCC라고 한다.

어떻게 SCC를 찾지?

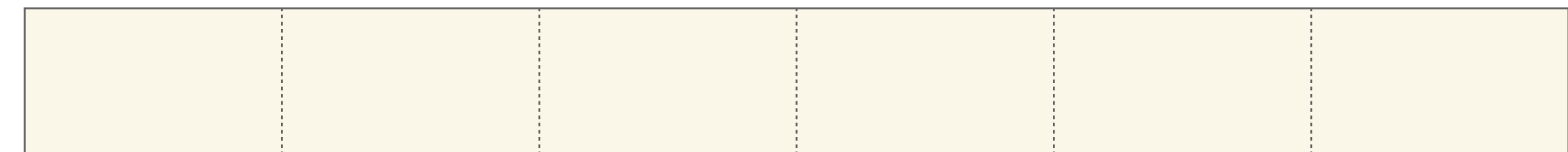


1. 간선의 방향을 뒤집은 **Reverse 그래프** 준비!

어떻게 SCC를 찾지?



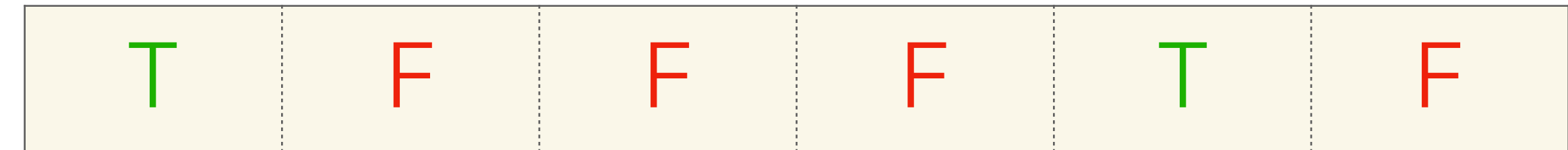
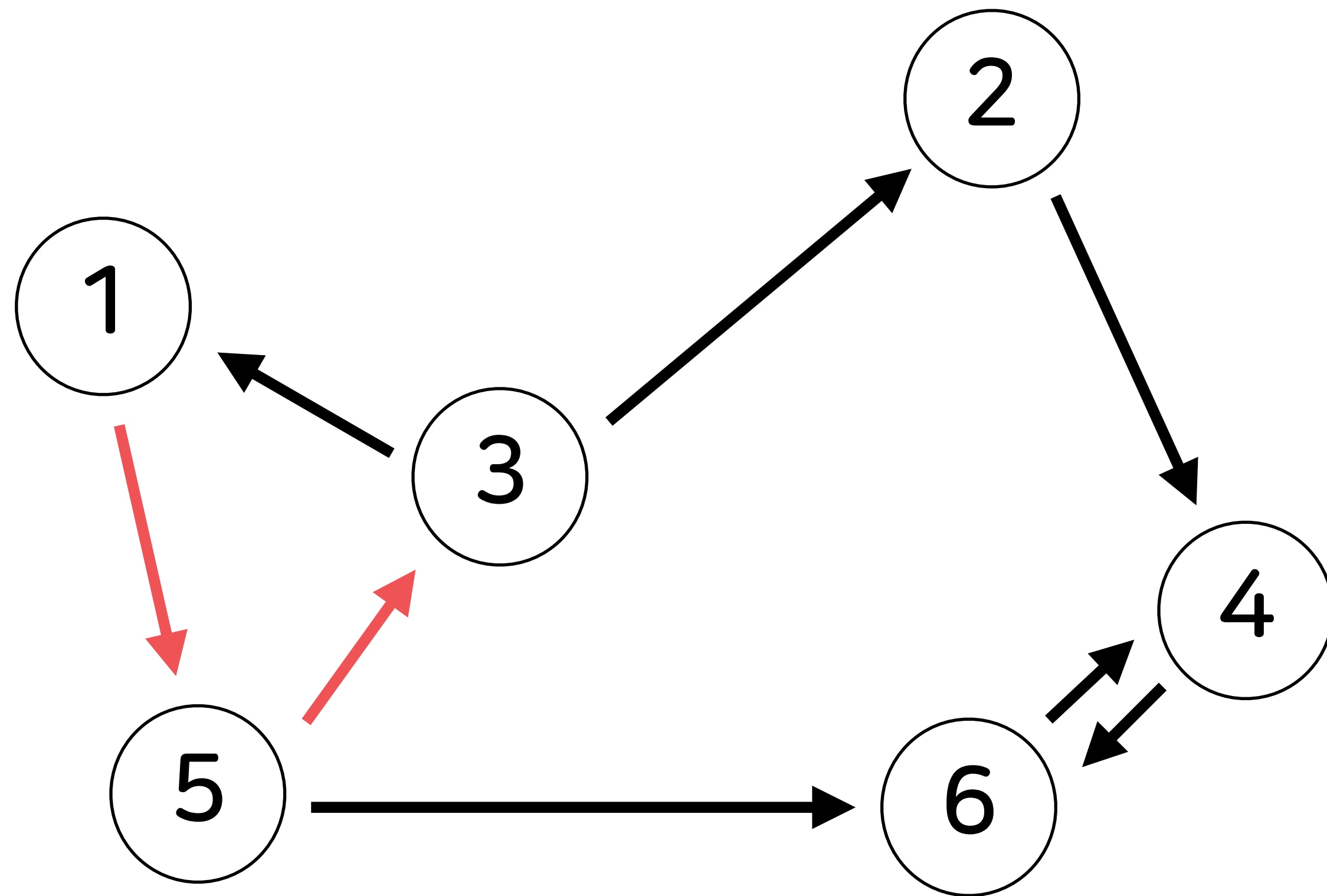
visited



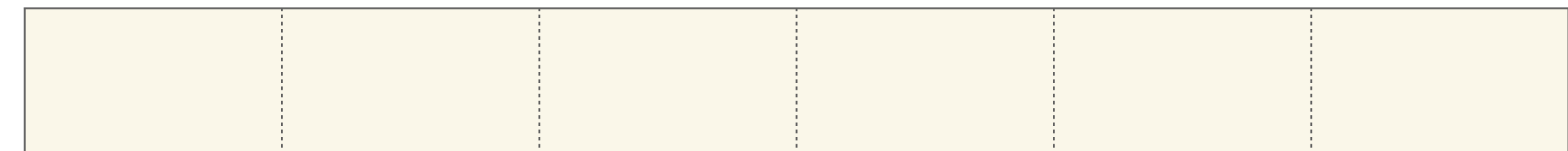
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



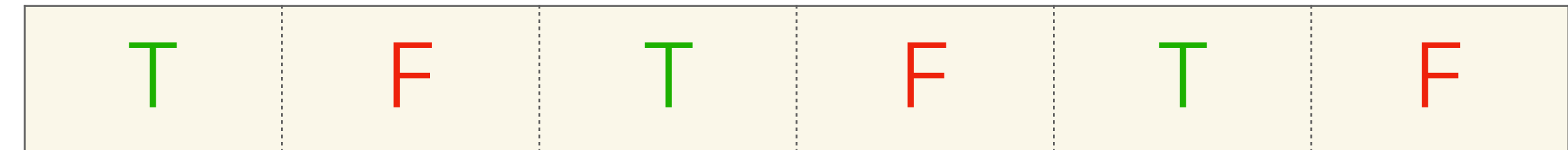
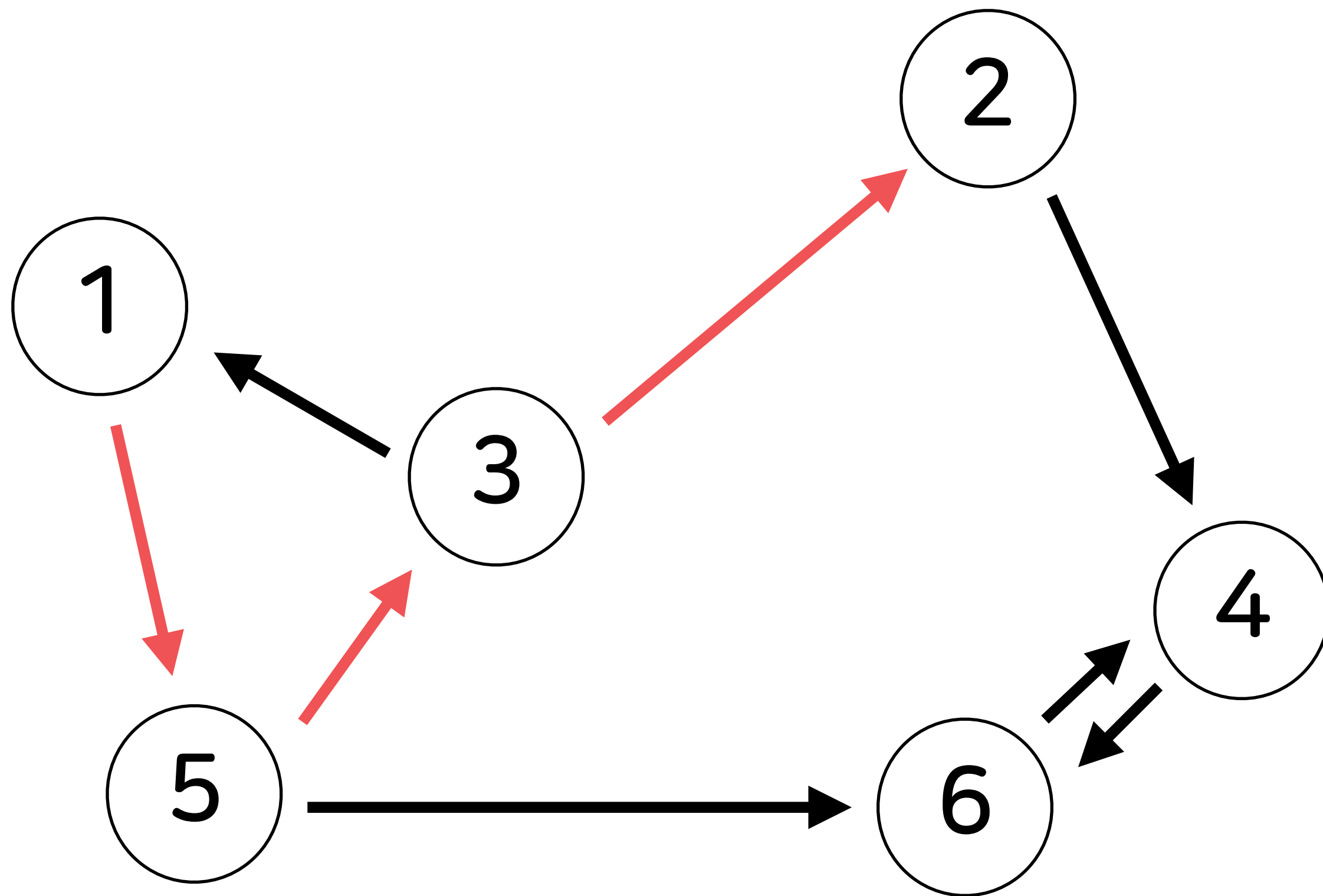
visited



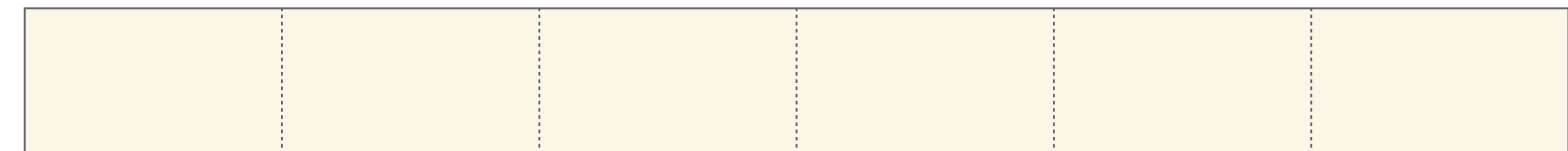
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



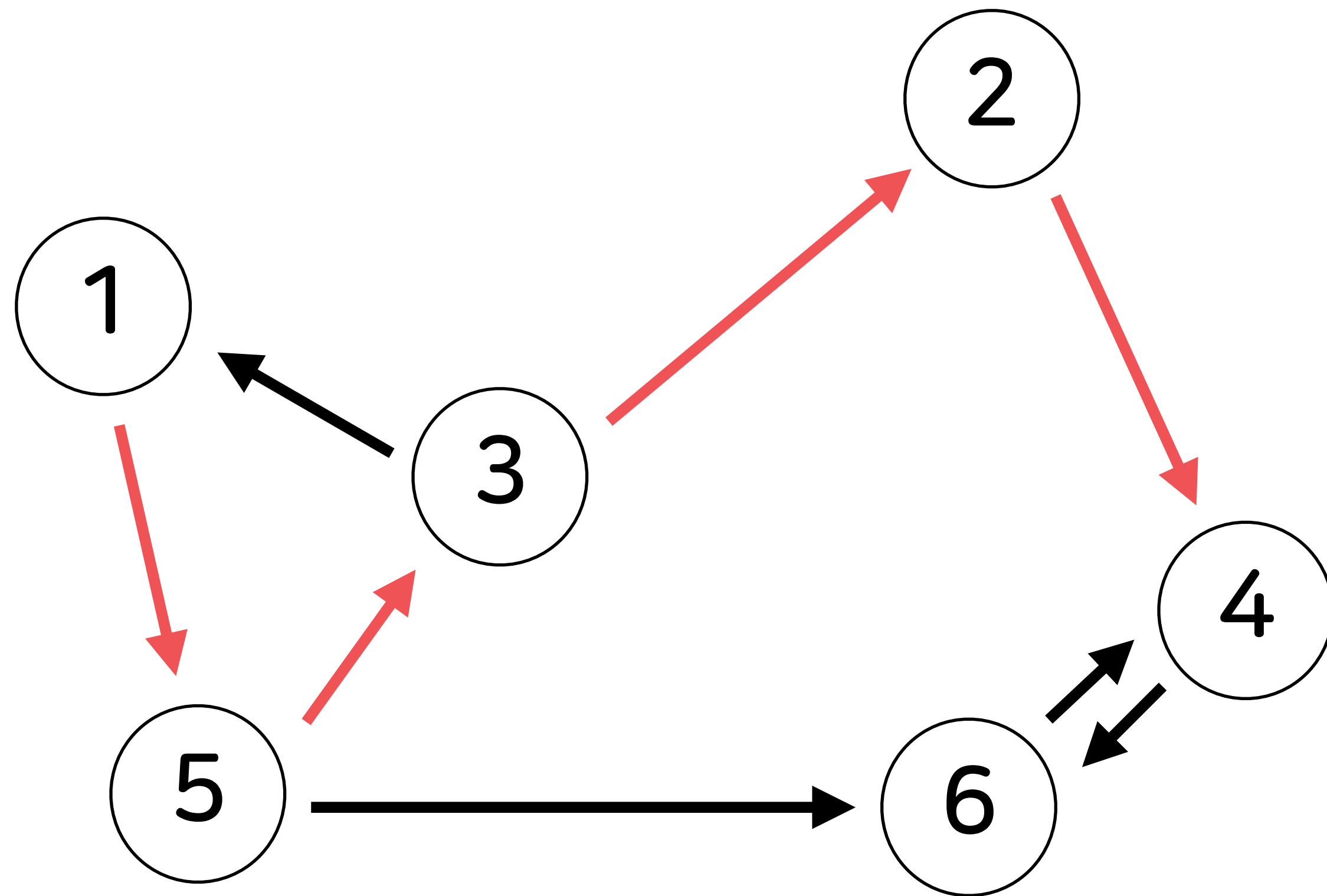
visited



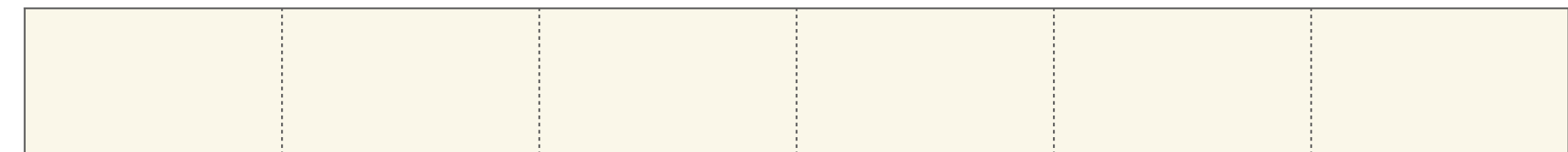
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



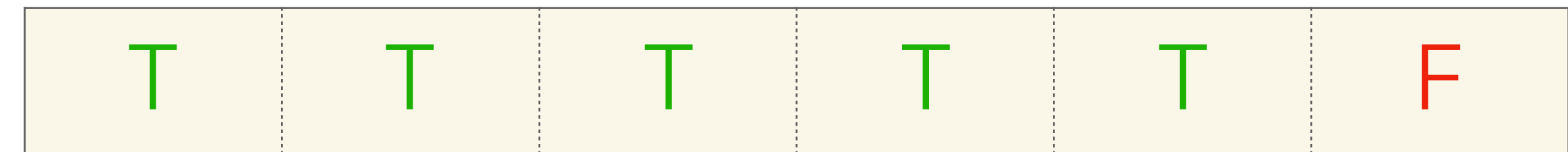
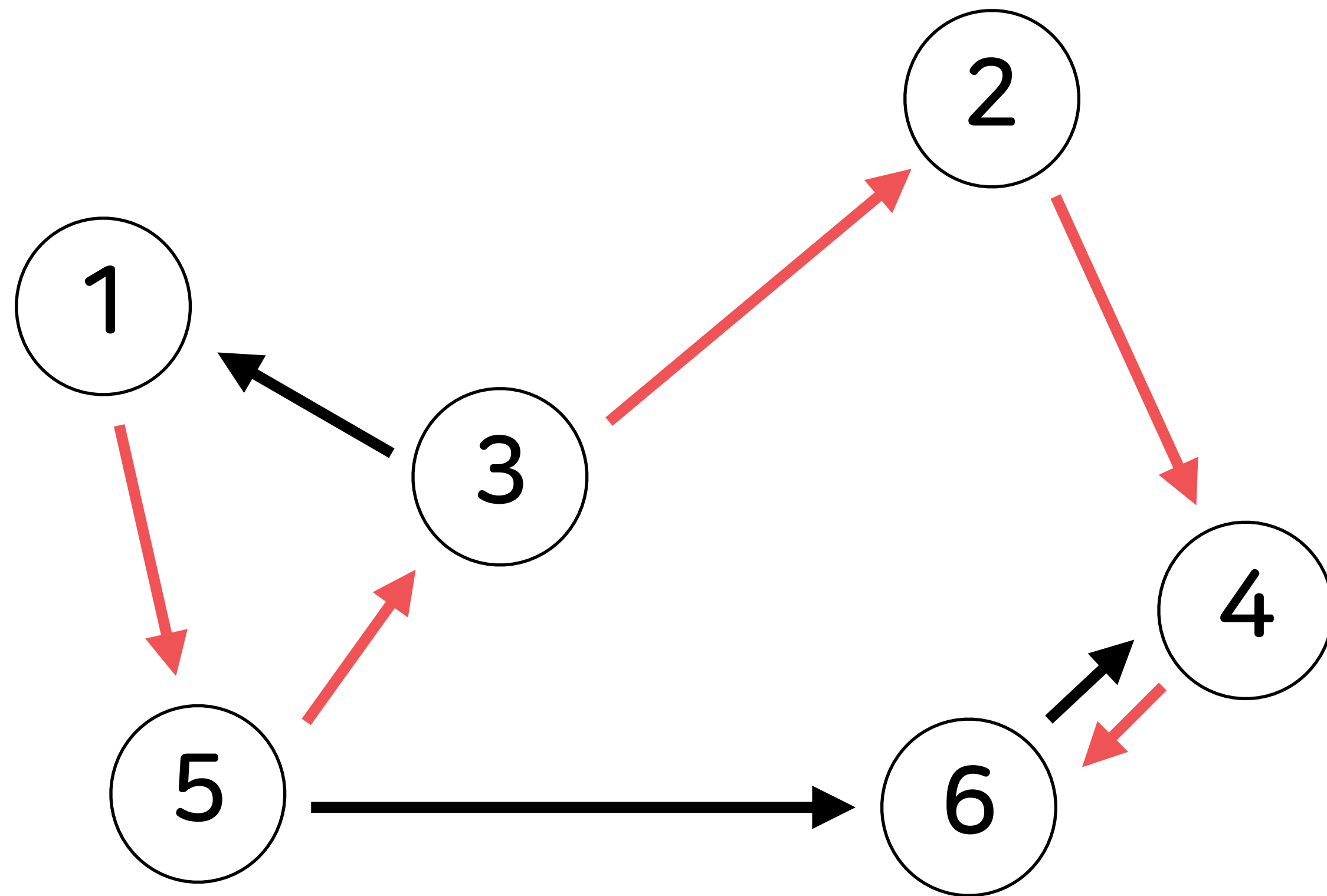
visited



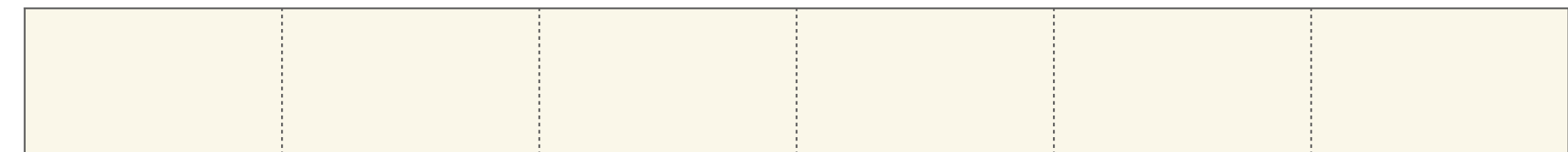
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



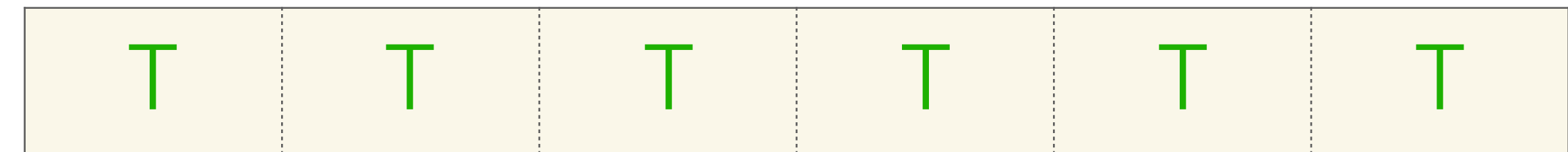
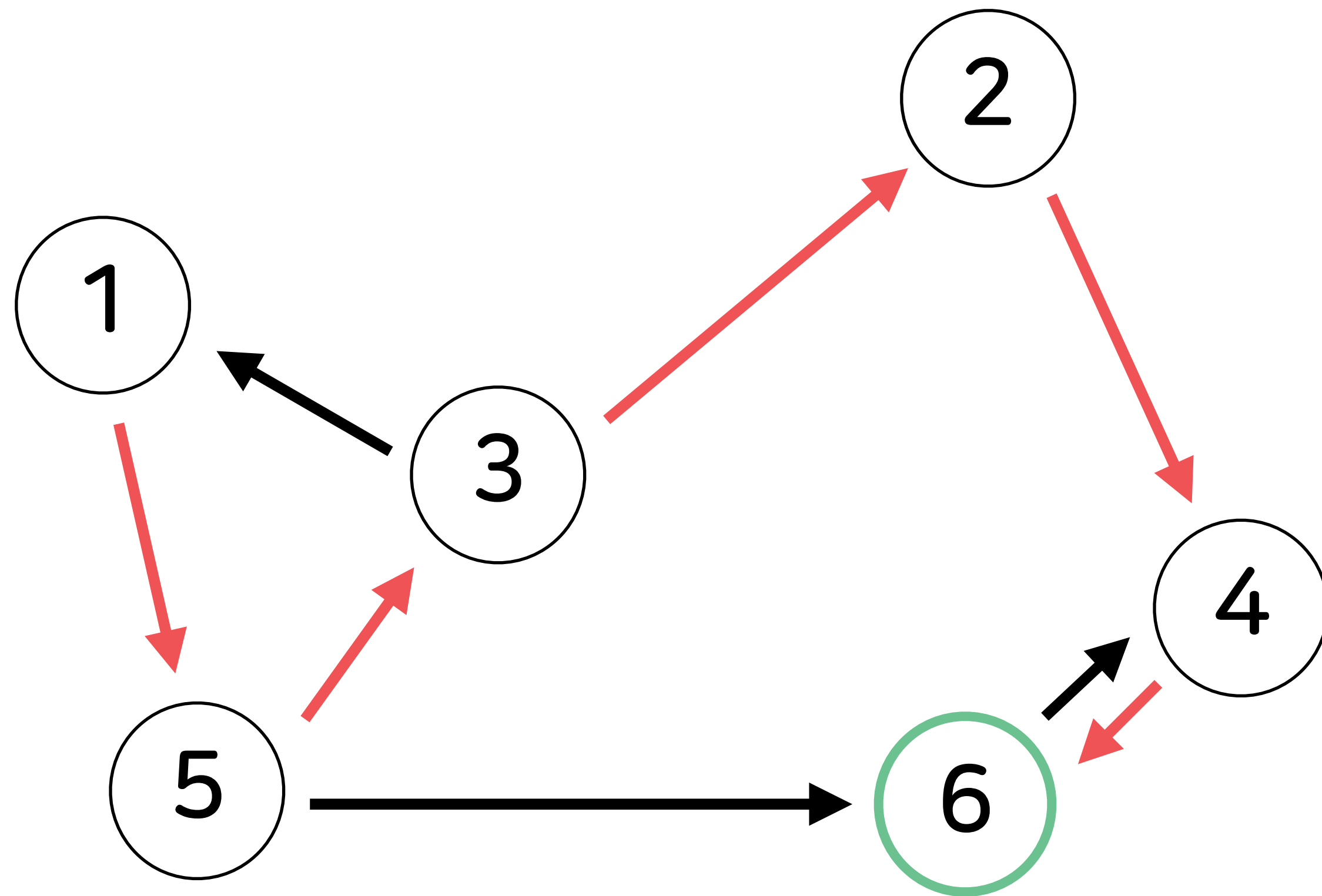
visited



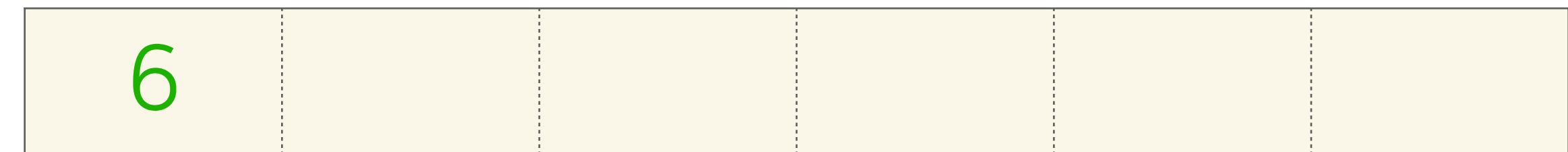
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



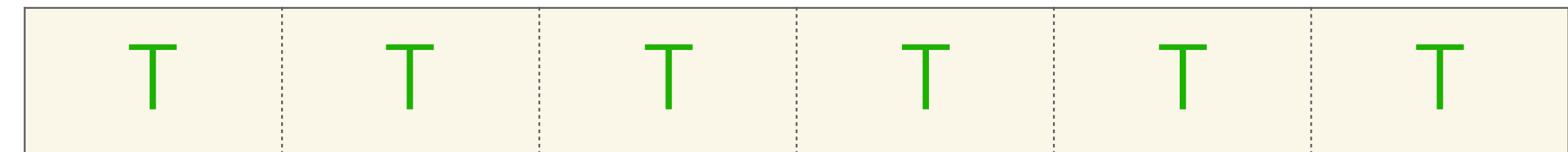
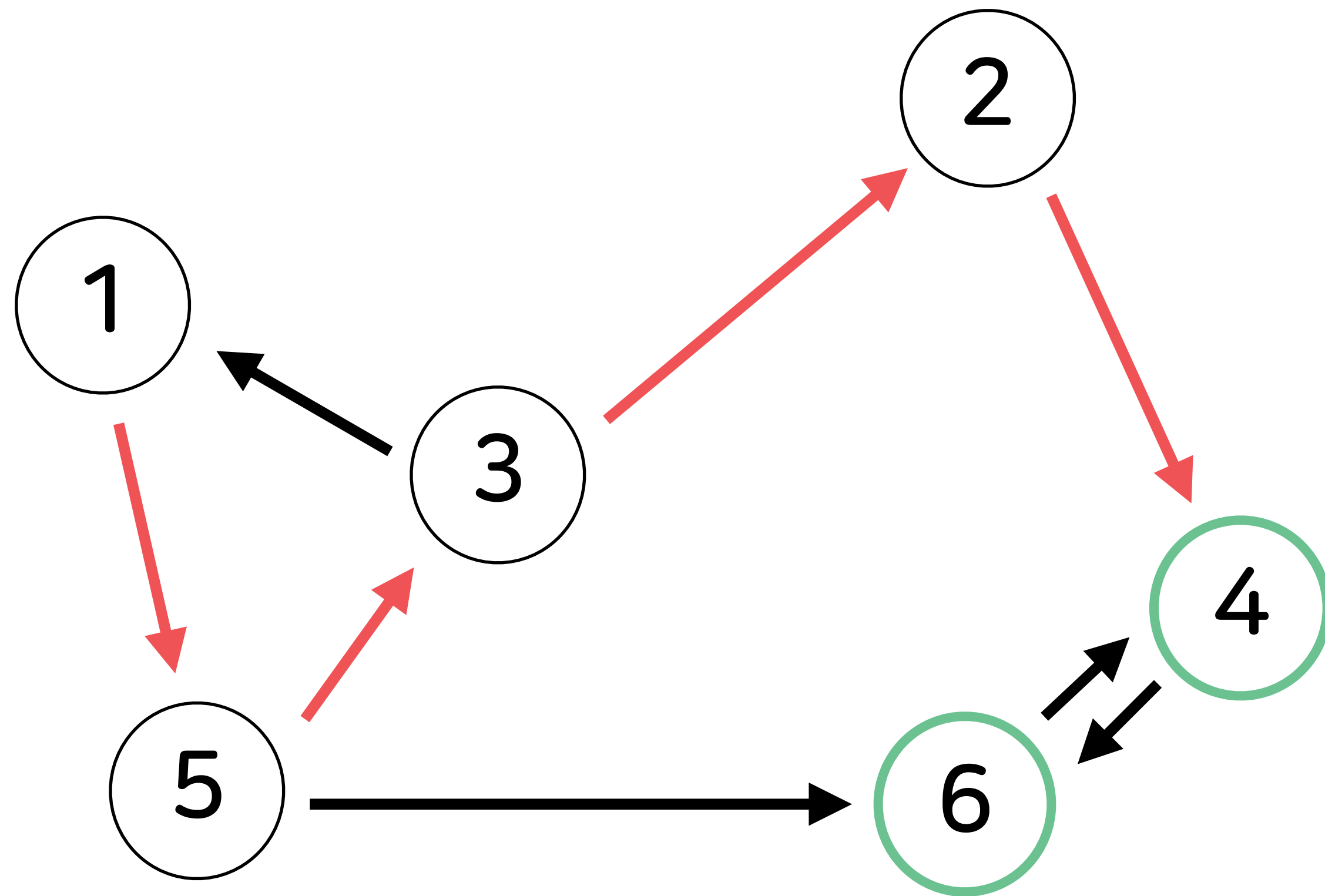
visited



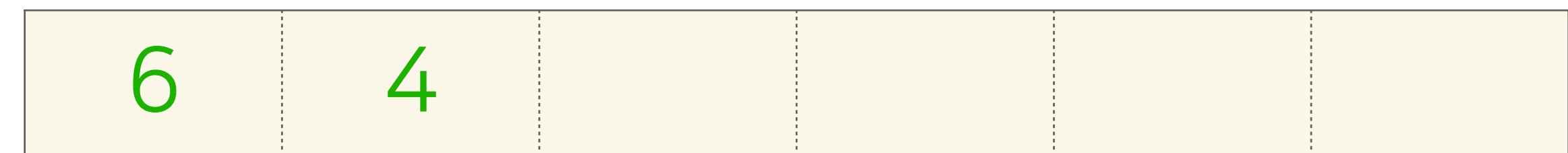
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



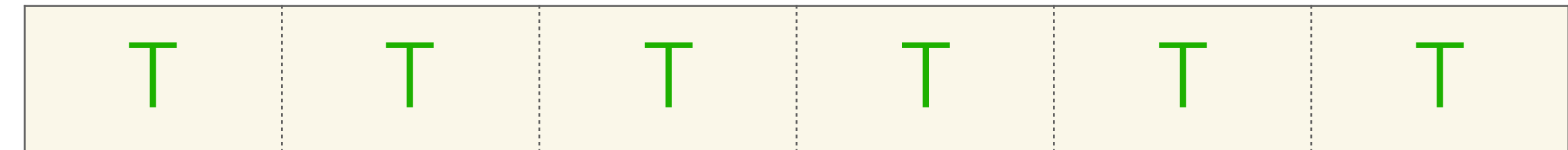
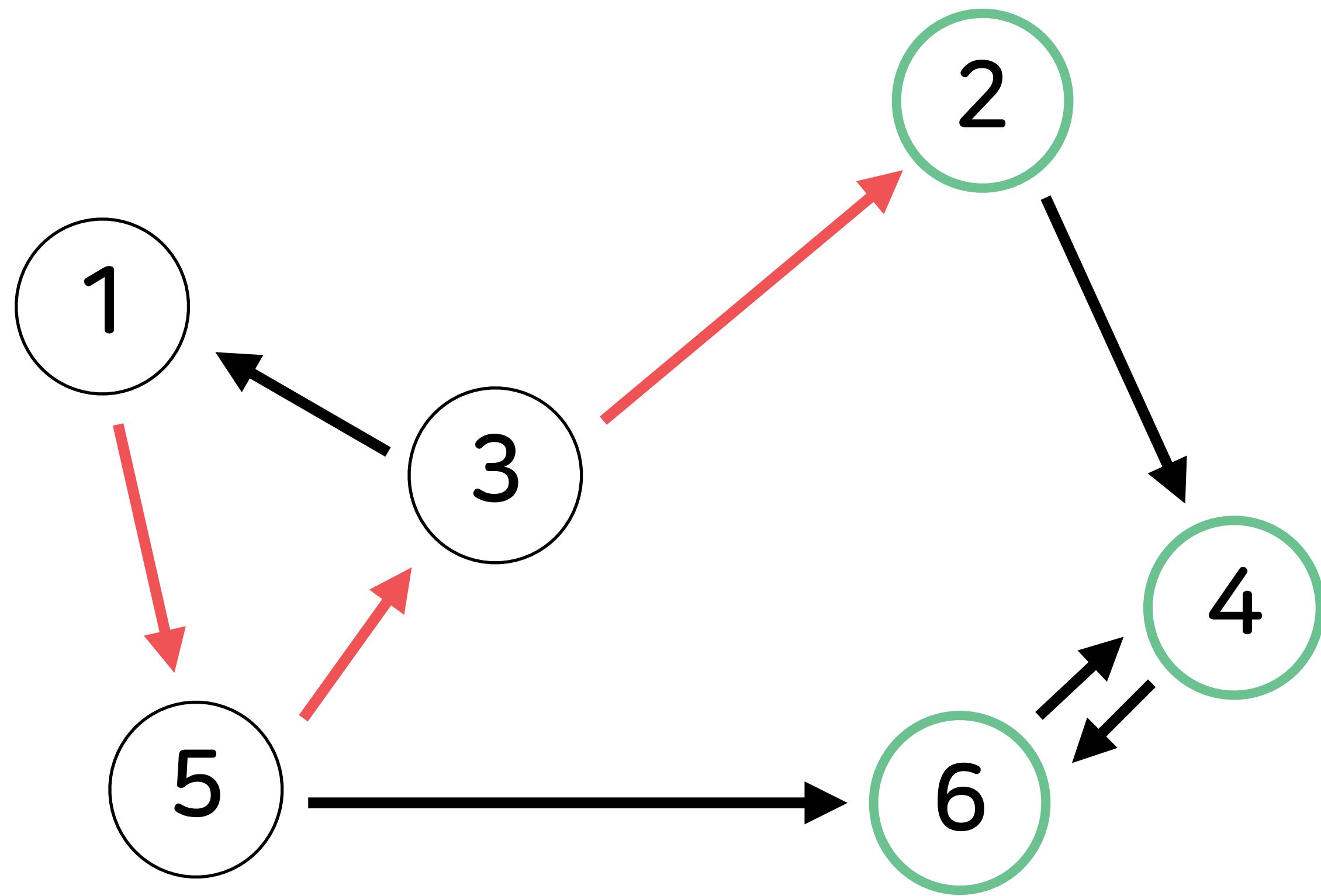
visited



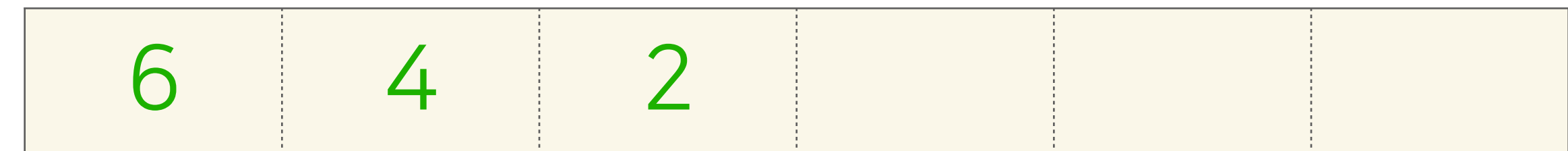
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



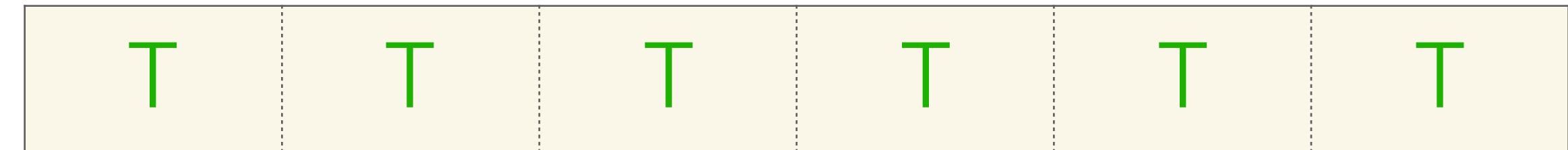
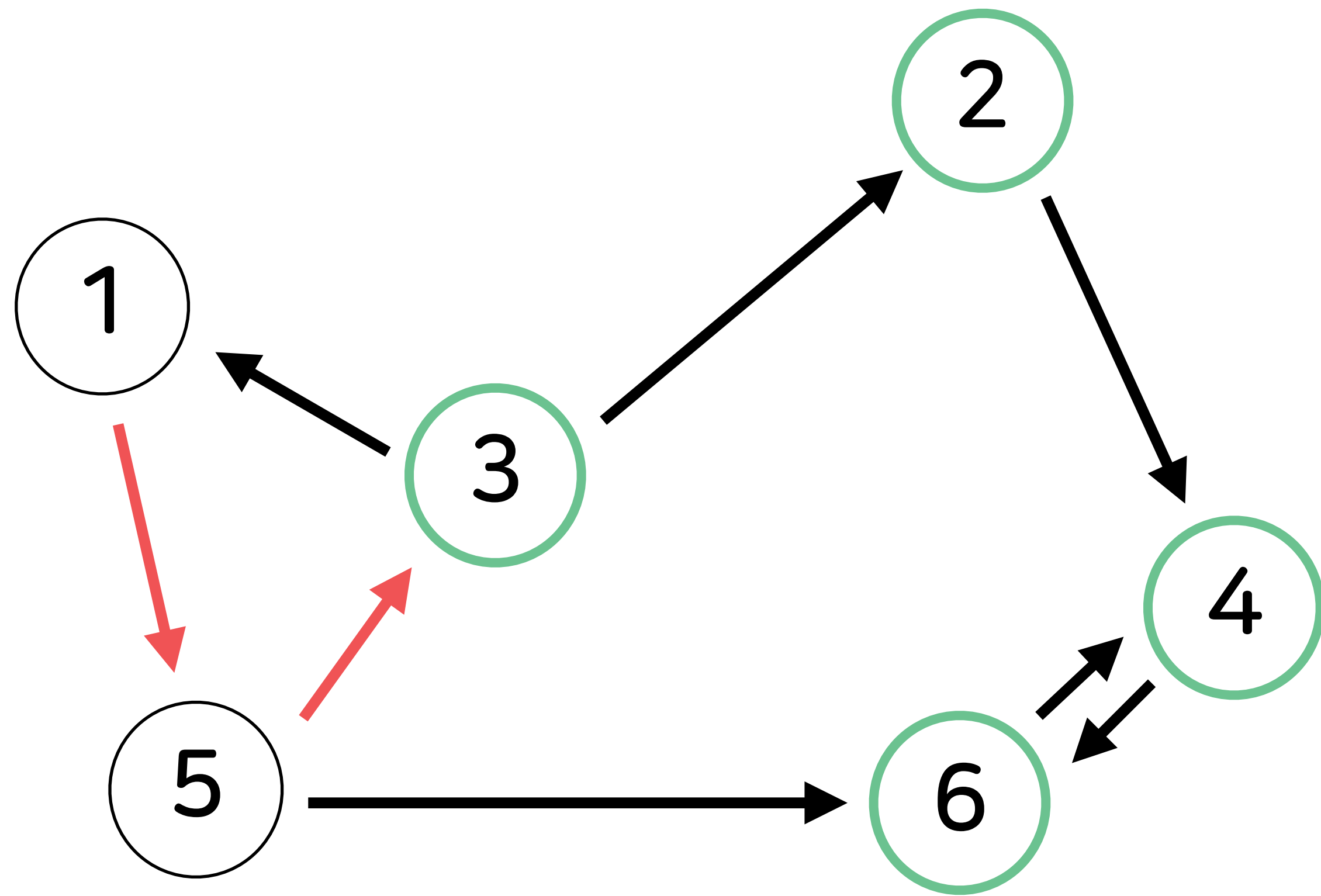
visited



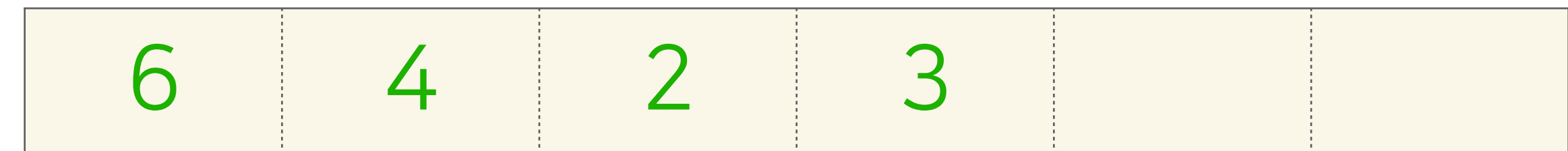
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



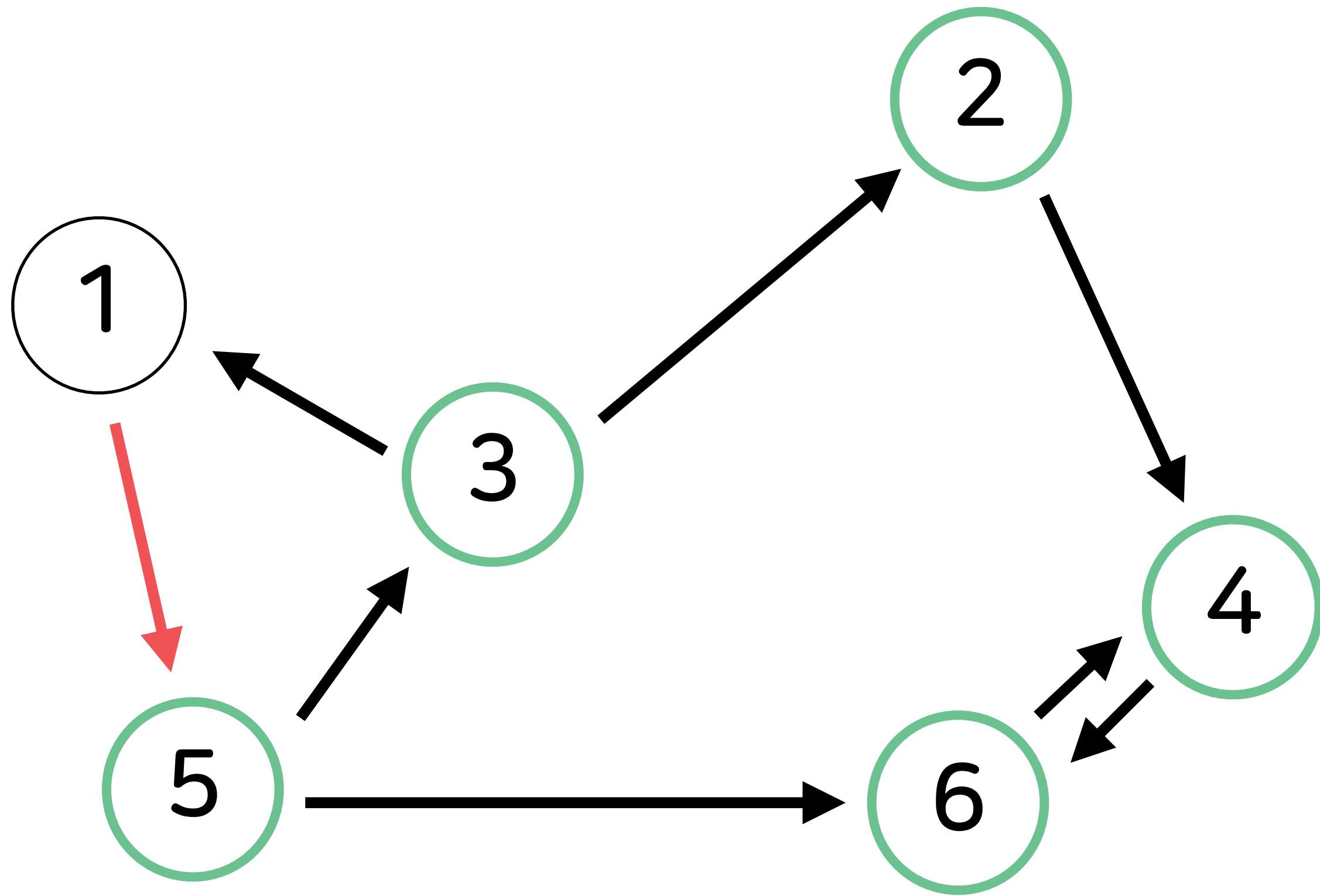
visited



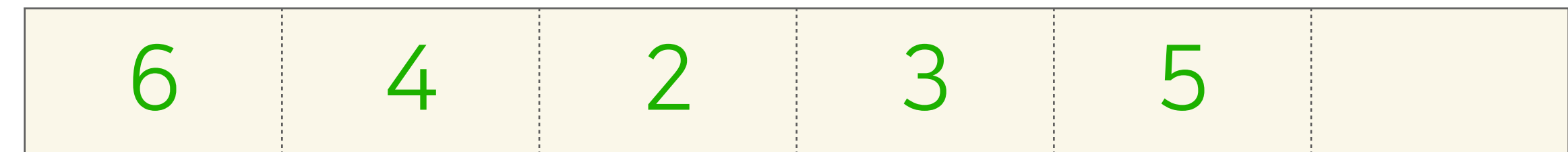
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



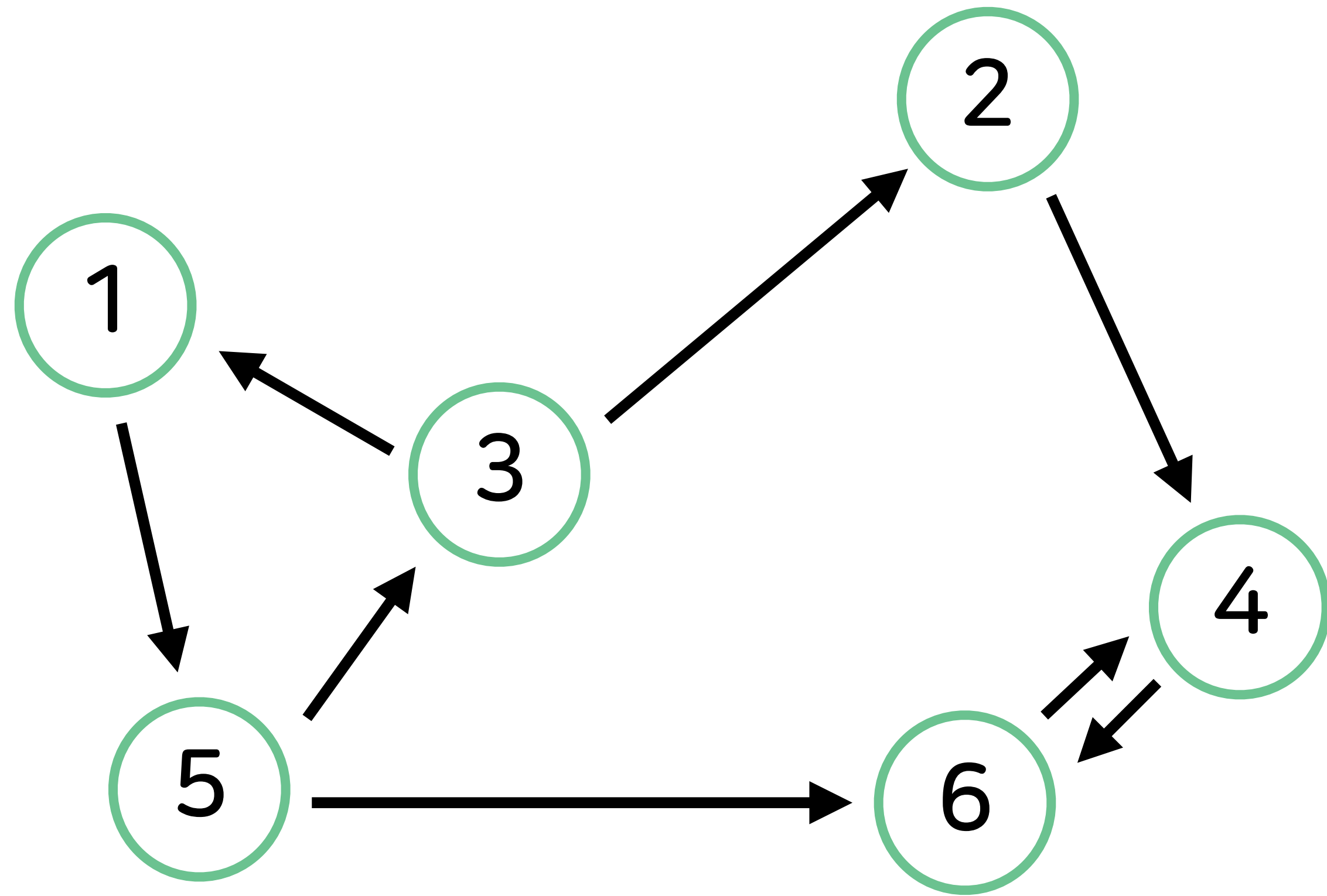
visited



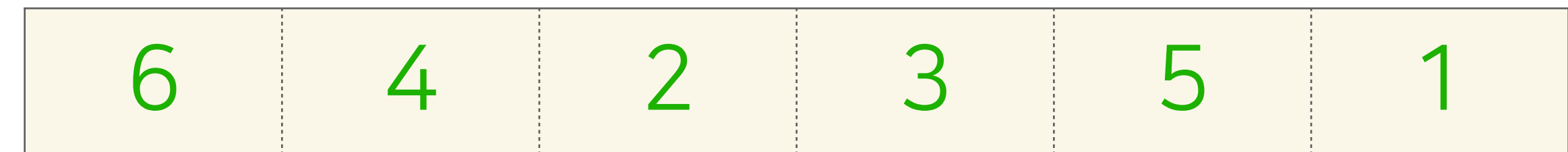
Check (Stack)

2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



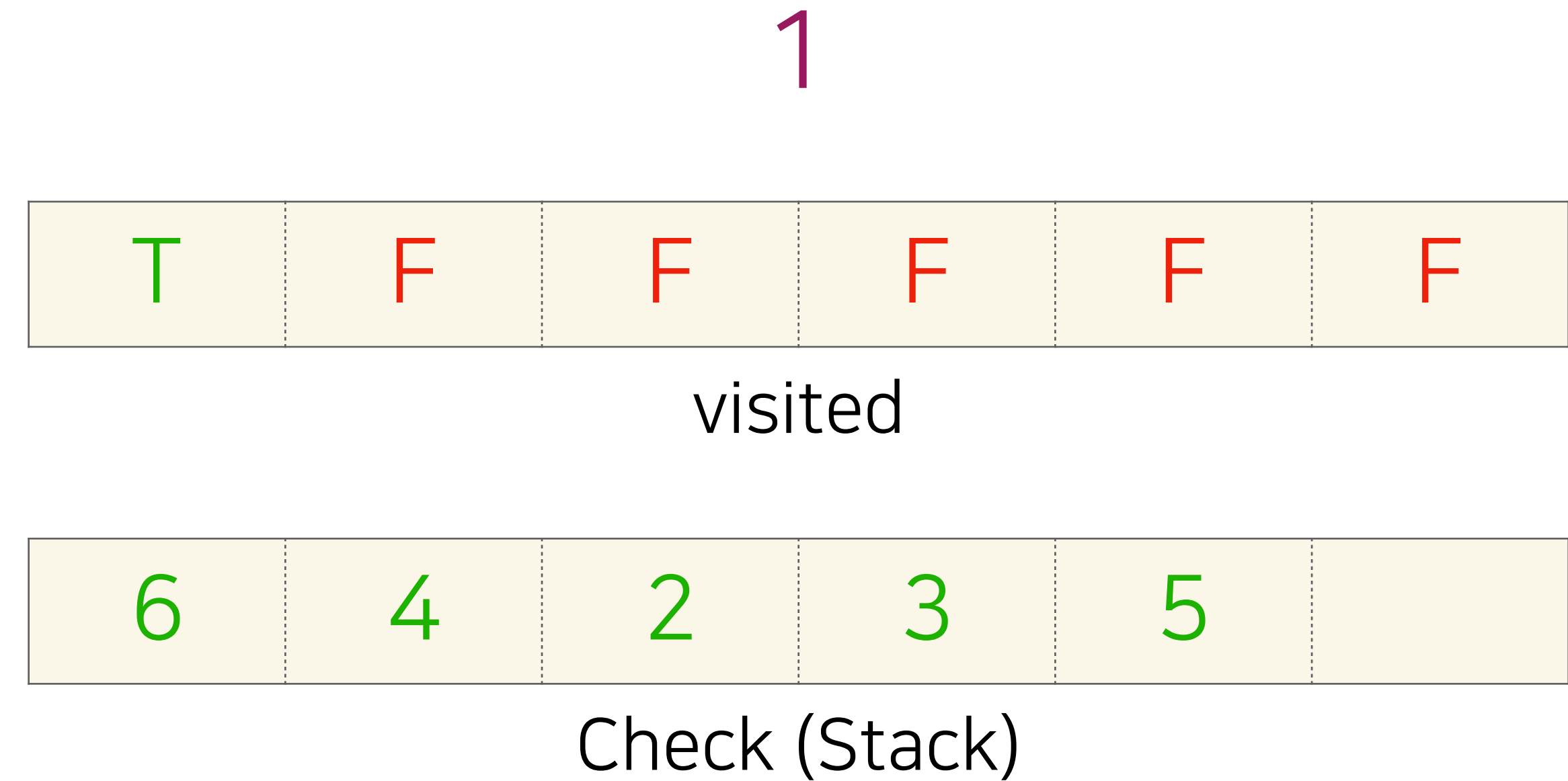
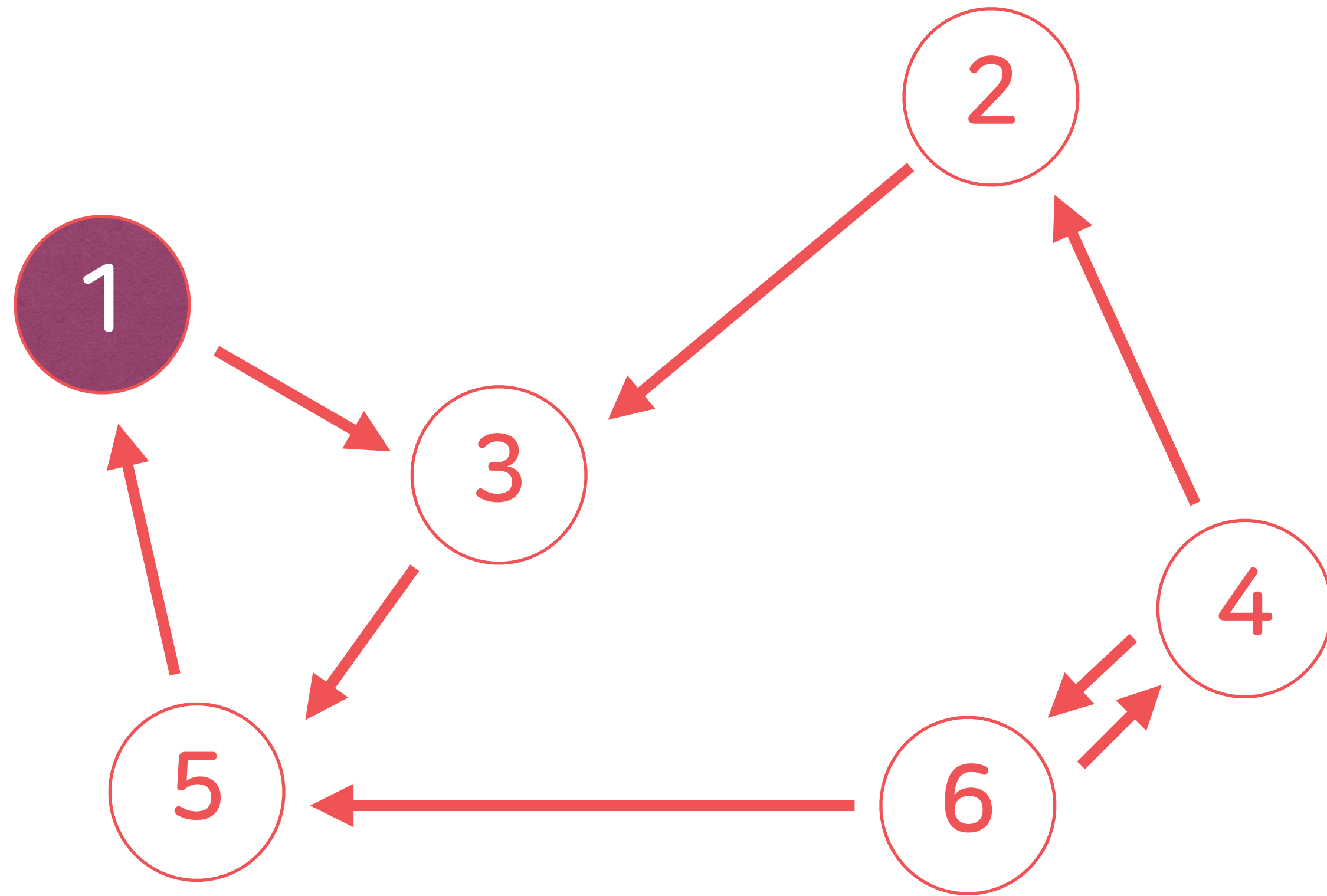
visited



Check (Stack)

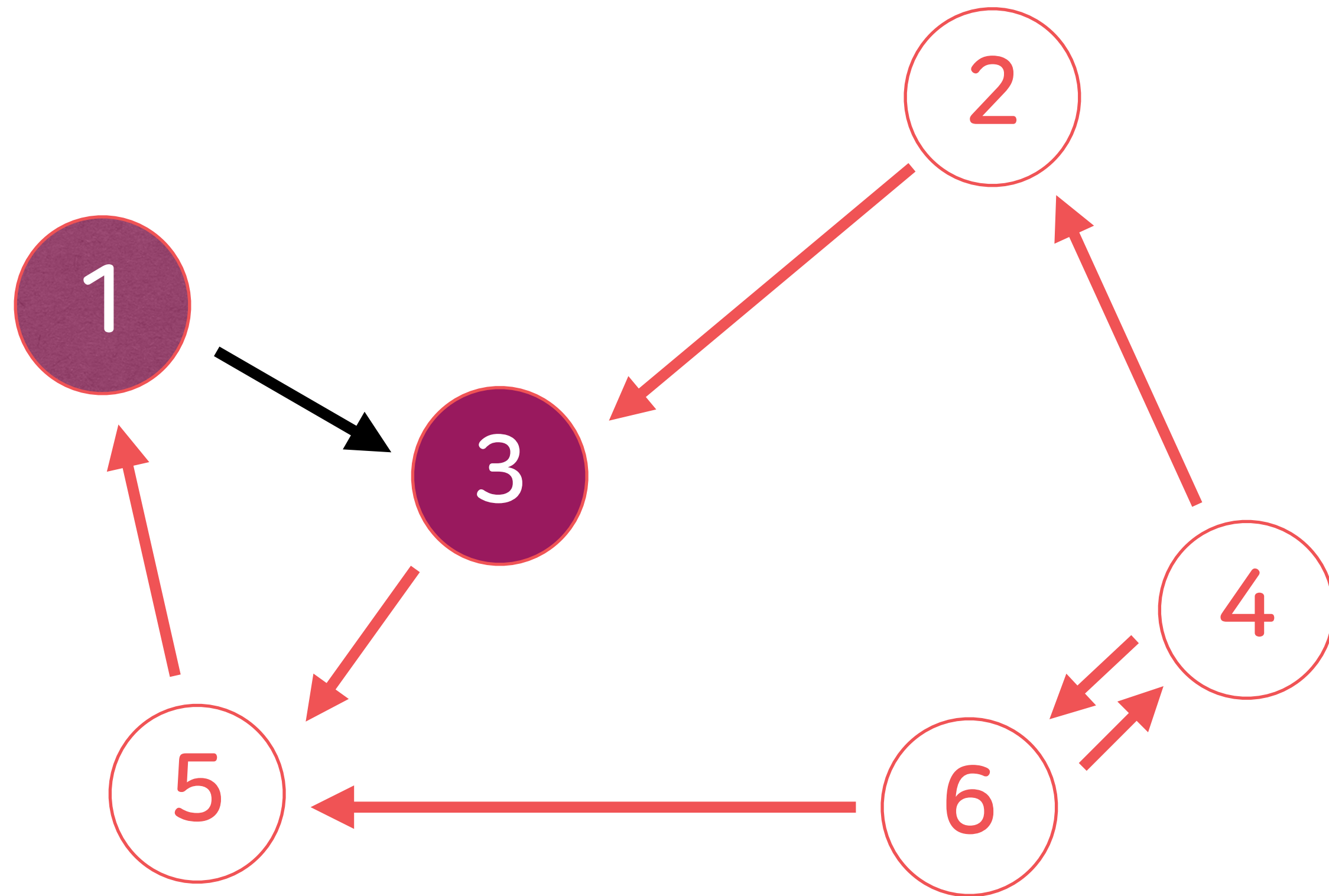
2. 정방향 그래프에 대해 DFS 진행 → 끝에 다다르면 **Check.push()**

어떻게 SCC를 찾지?



3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?



1 3

T	F	T	F	F	F
---	---	---	---	---	---

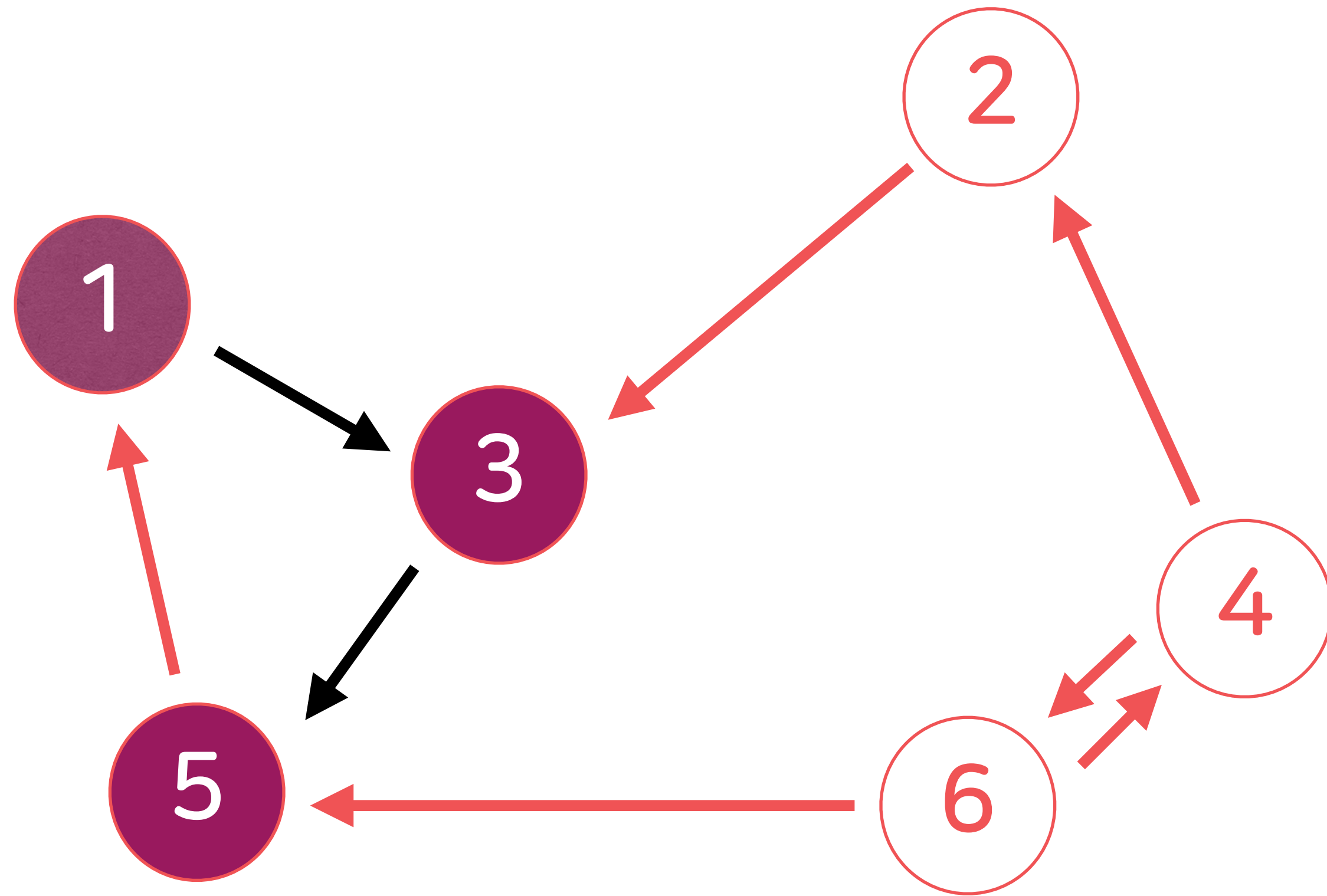
visited

6	4	2	3	5	
---	---	---	---	---	--

Check (Stack)

3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?



1 3 5

T	F	T	F	T	F
---	---	---	---	---	---

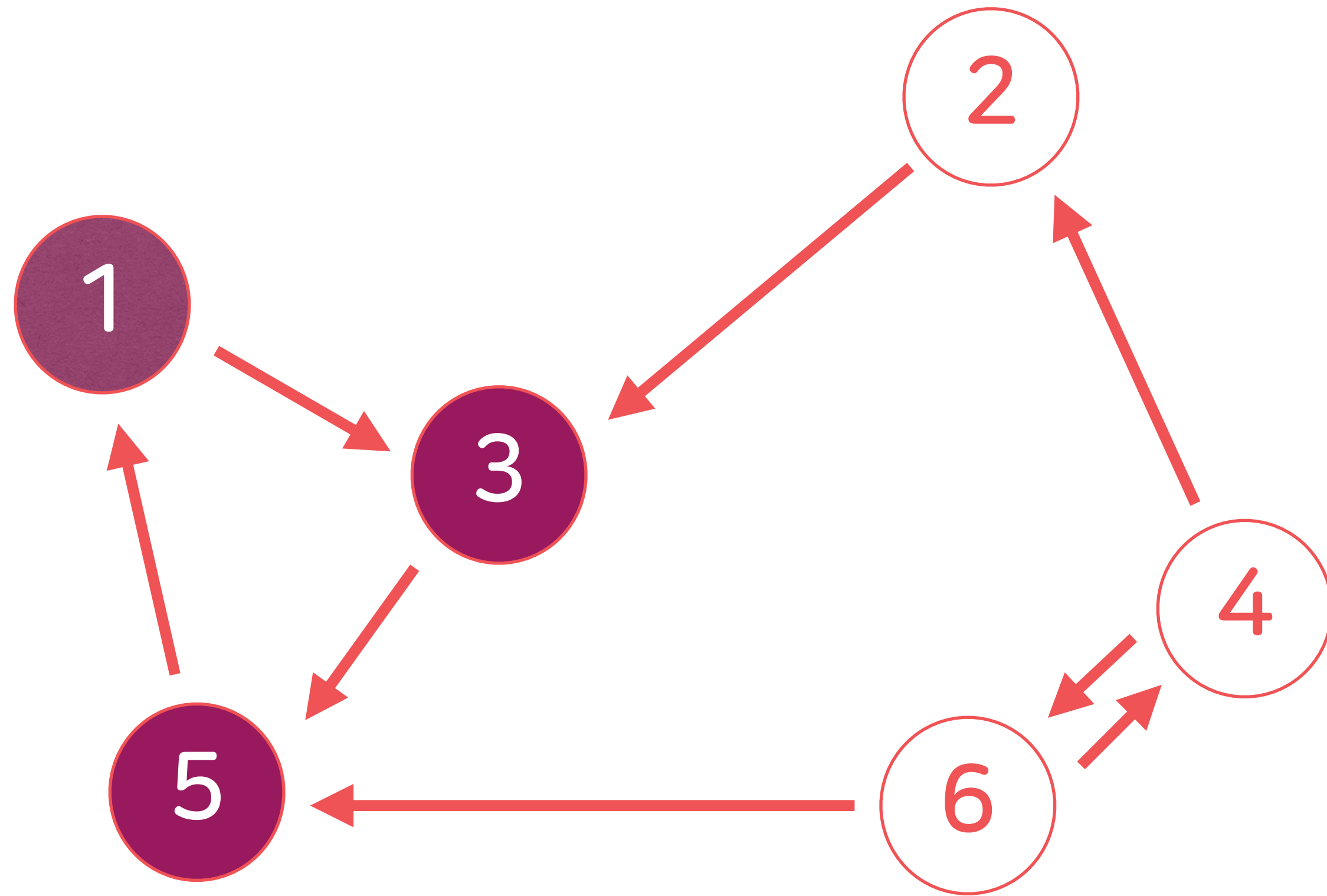
visited

6	4	2	3	5	
---	---	---	---	---	--

Check (Stack)

3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?



1 3 5

T	F	T	F	T	F
---	---	---	---	---	---

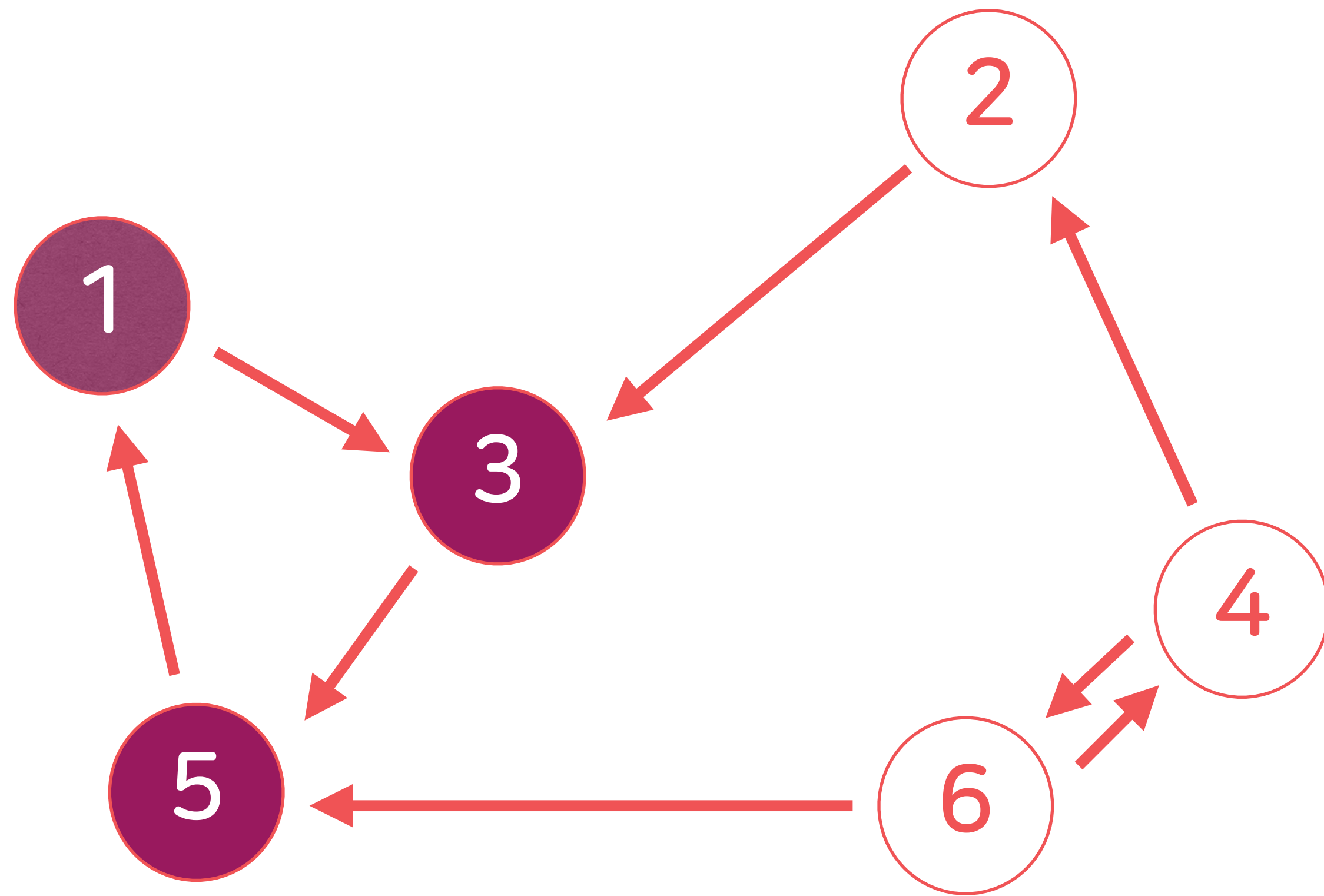
visited

6	4	2	3		
---	---	---	---	--	--

Check (Stack)

3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?



1 3 5

T	F	T	F	T	F
---	---	---	---	---	---

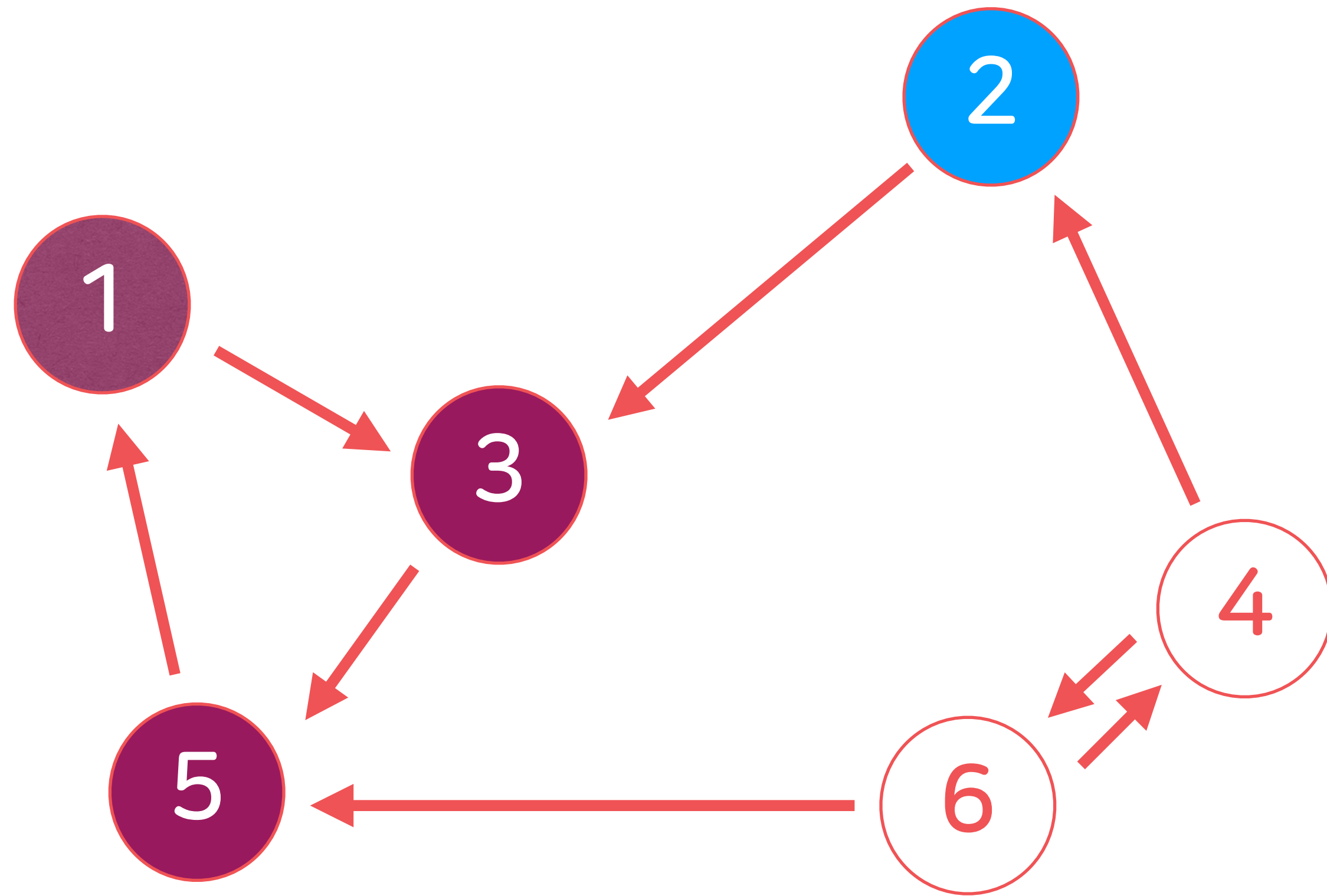
visited

6	4	2			
---	---	---	--	--	--

Check (Stack)

3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?



2

T	T	T	F	T	F
---	---	---	---	---	---

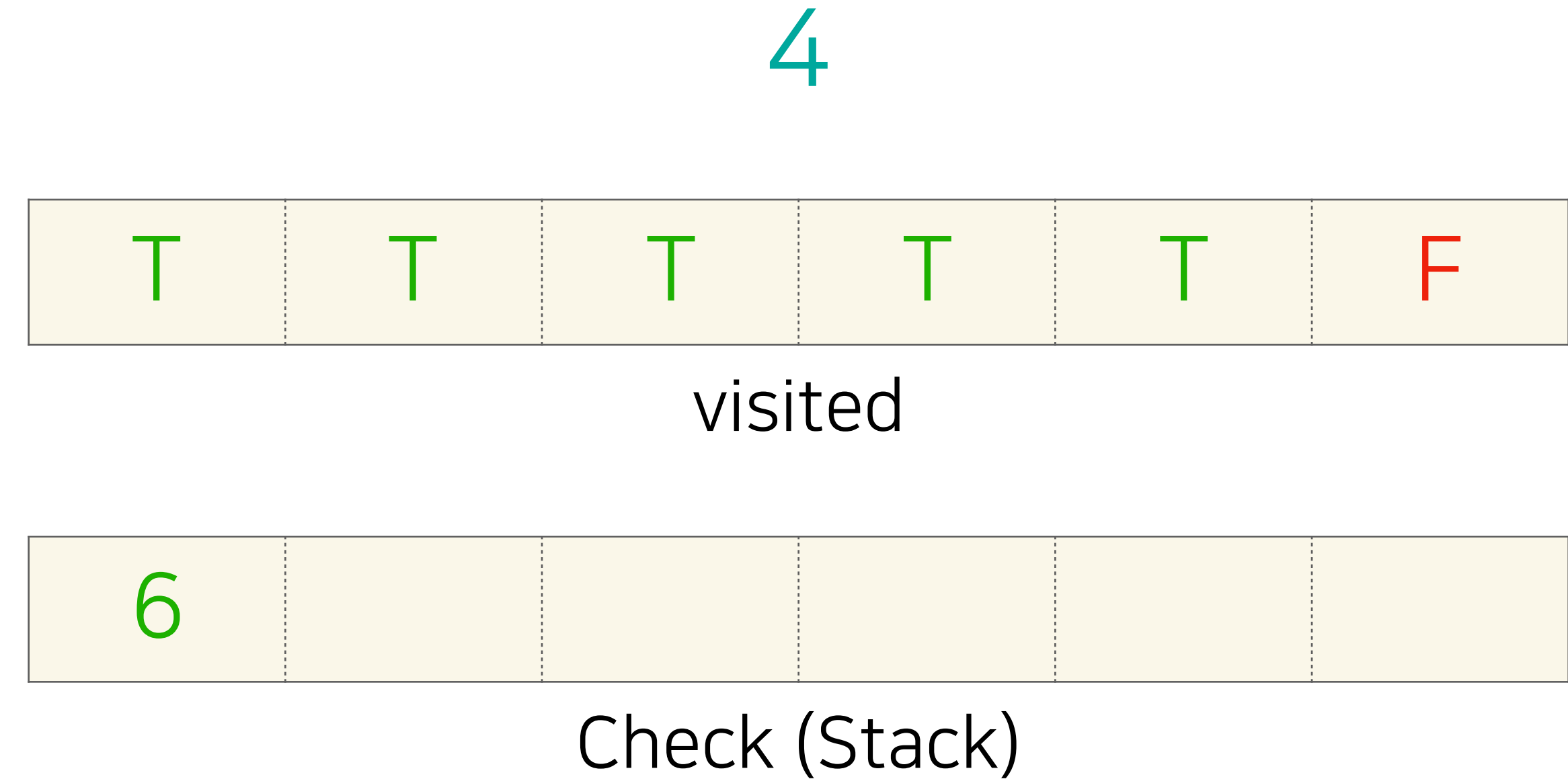
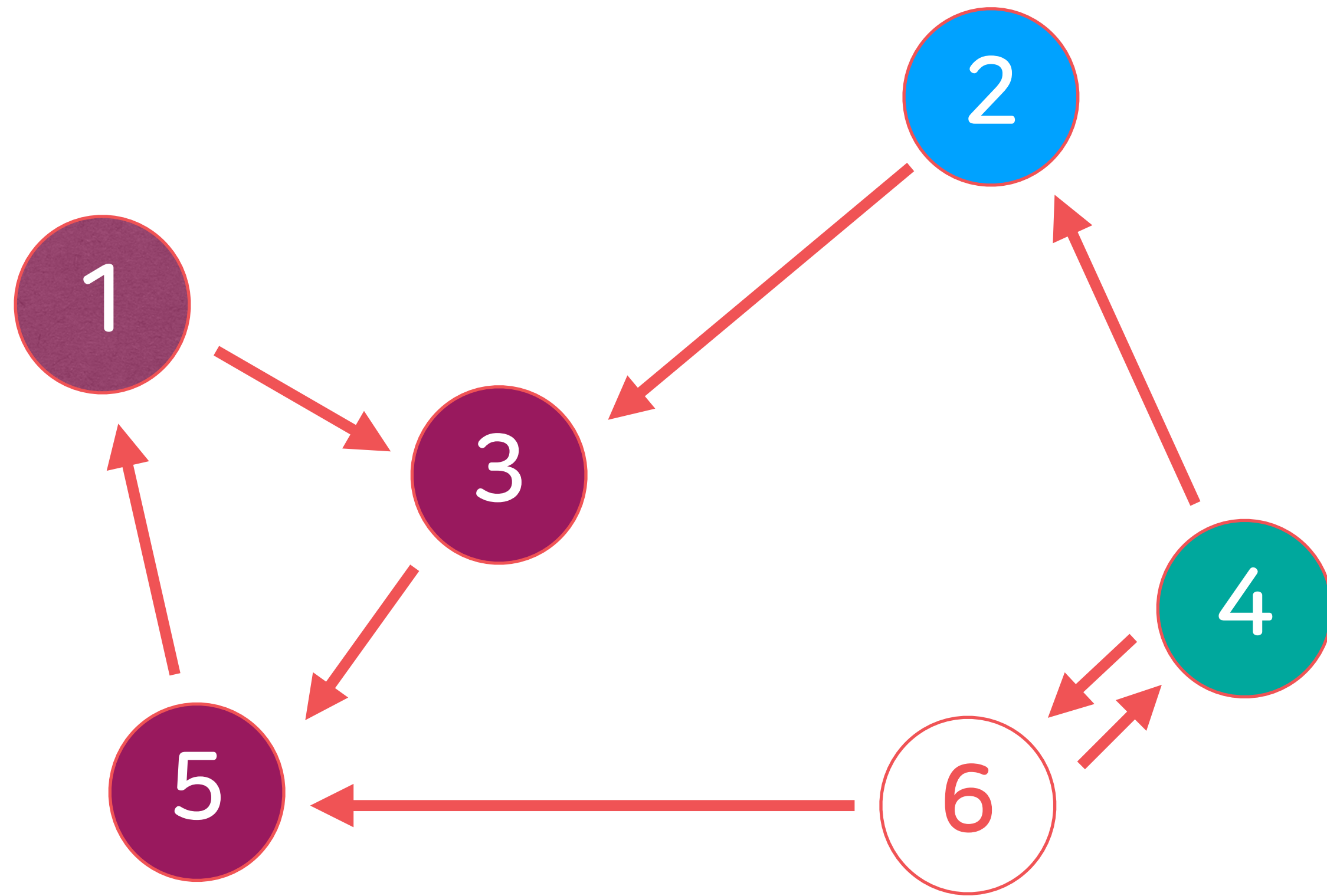
visited

6	4				
---	---	--	--	--	--

Check (Stack)

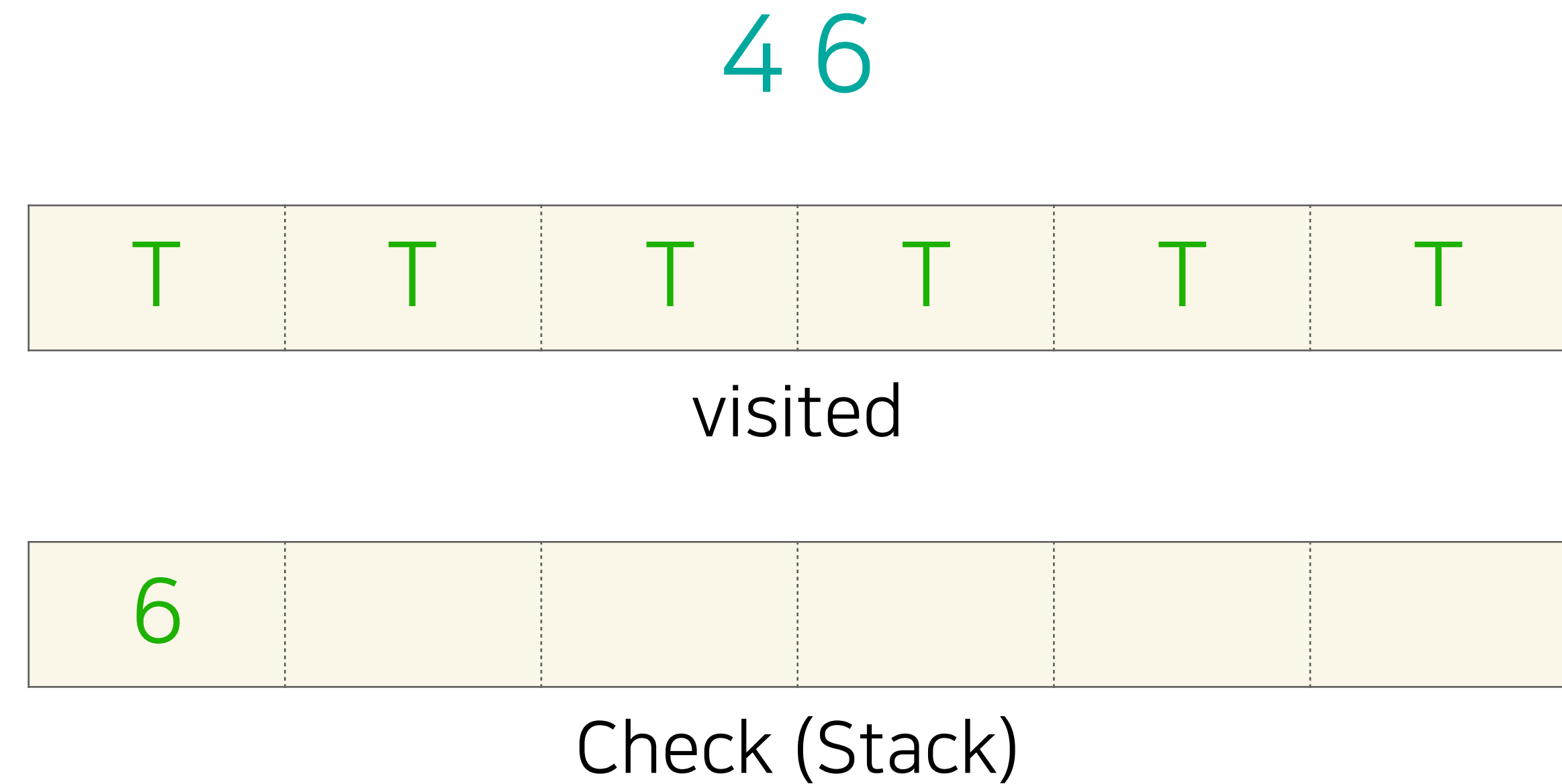
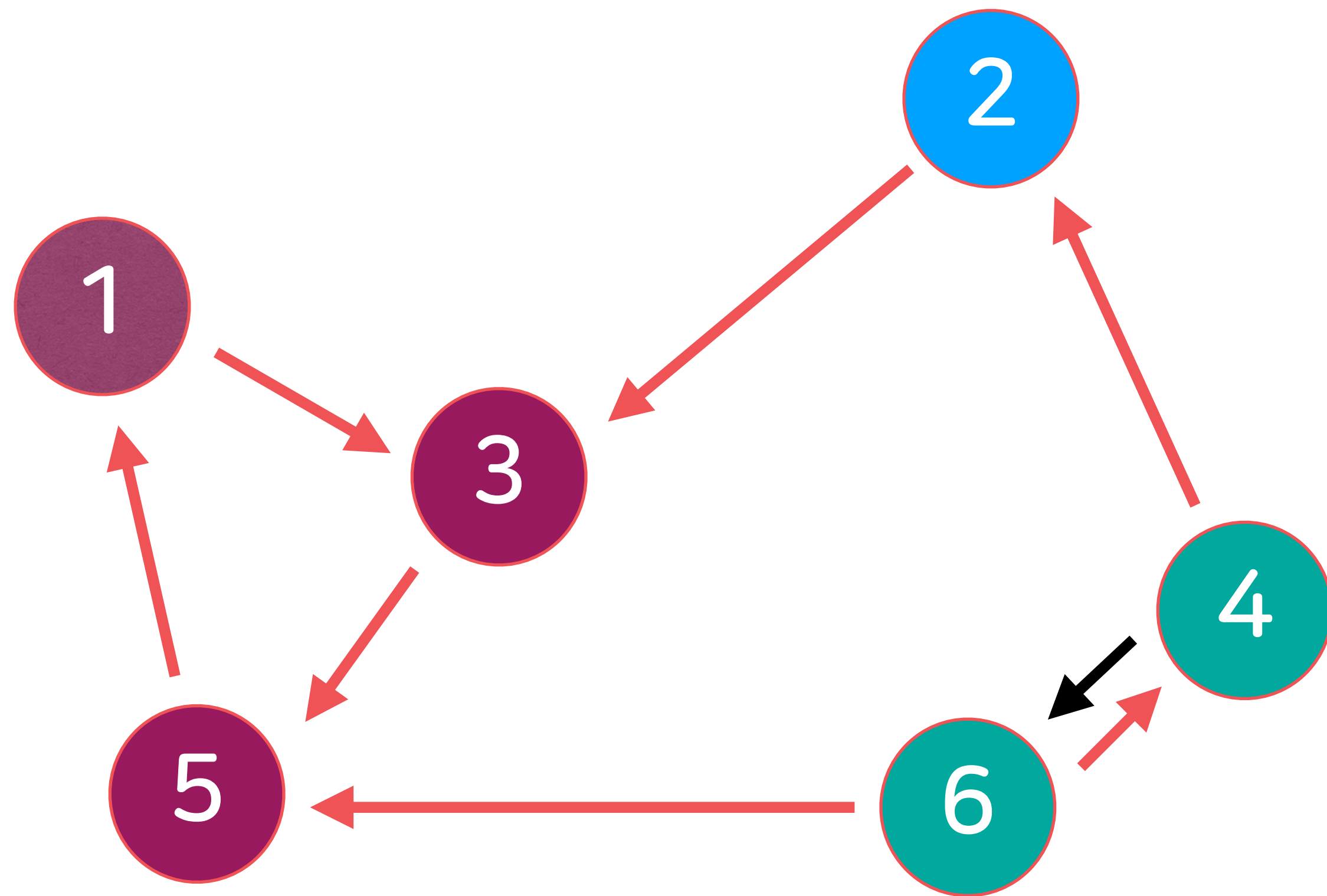
3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?



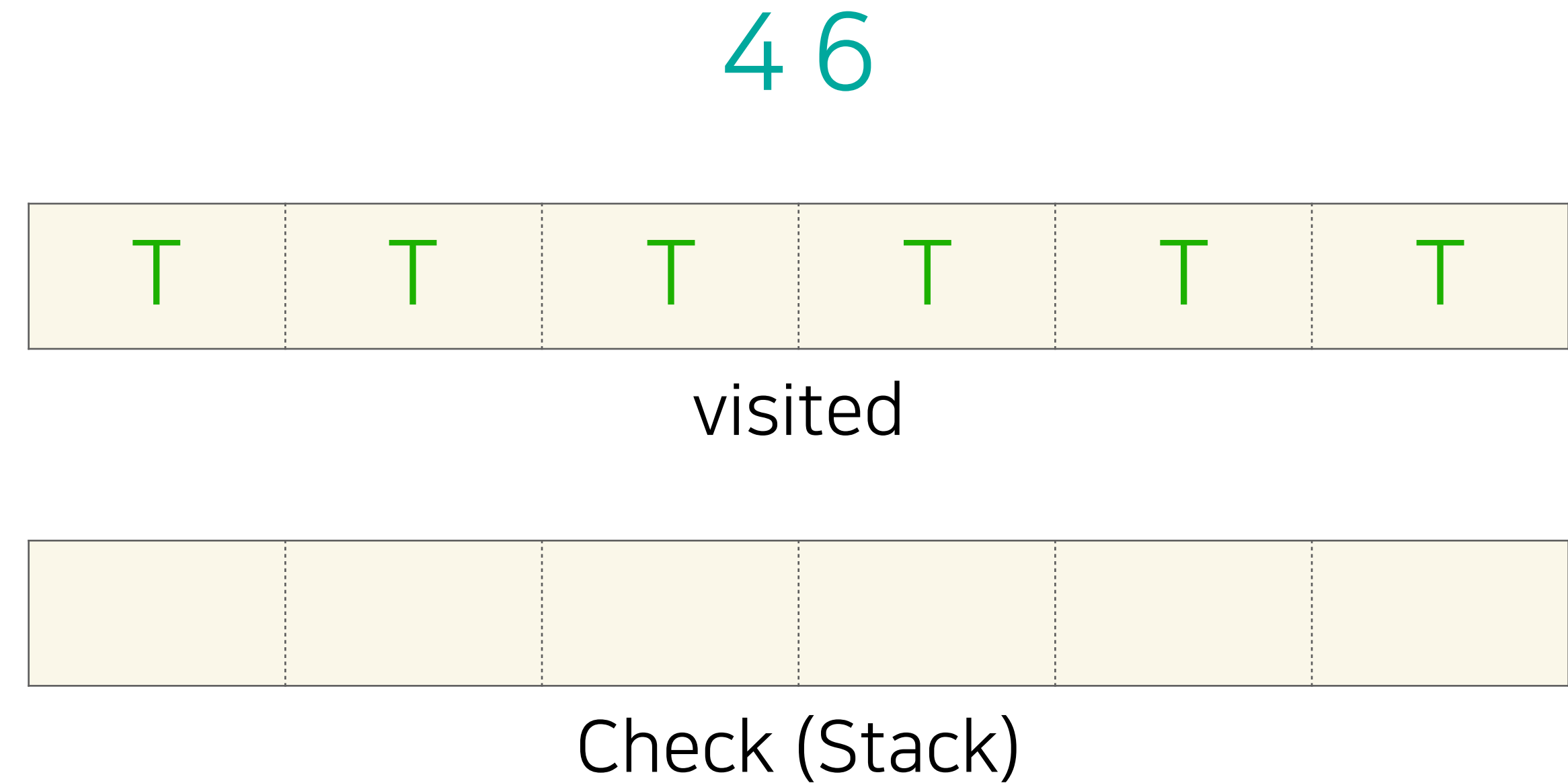
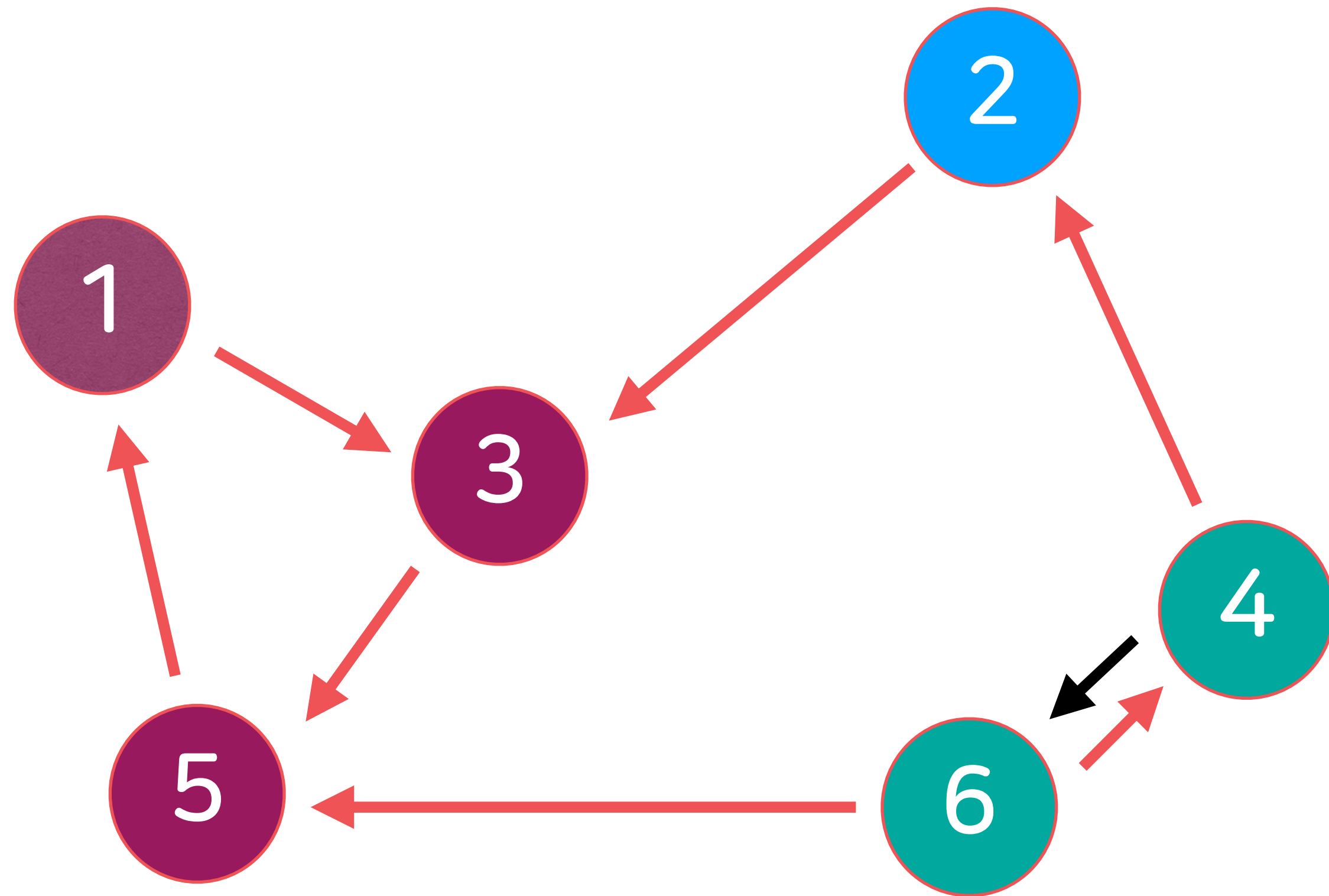
3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?



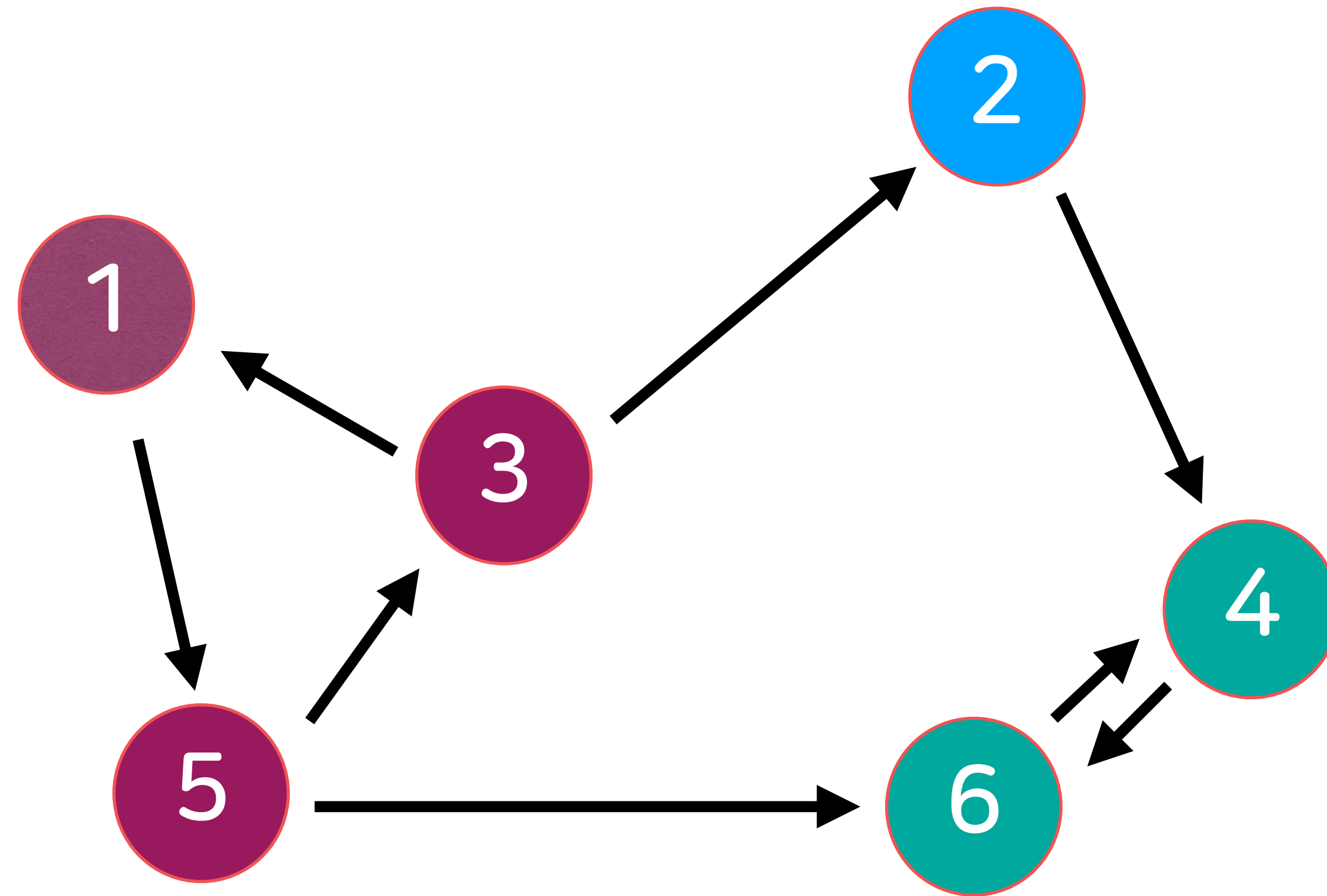
3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?



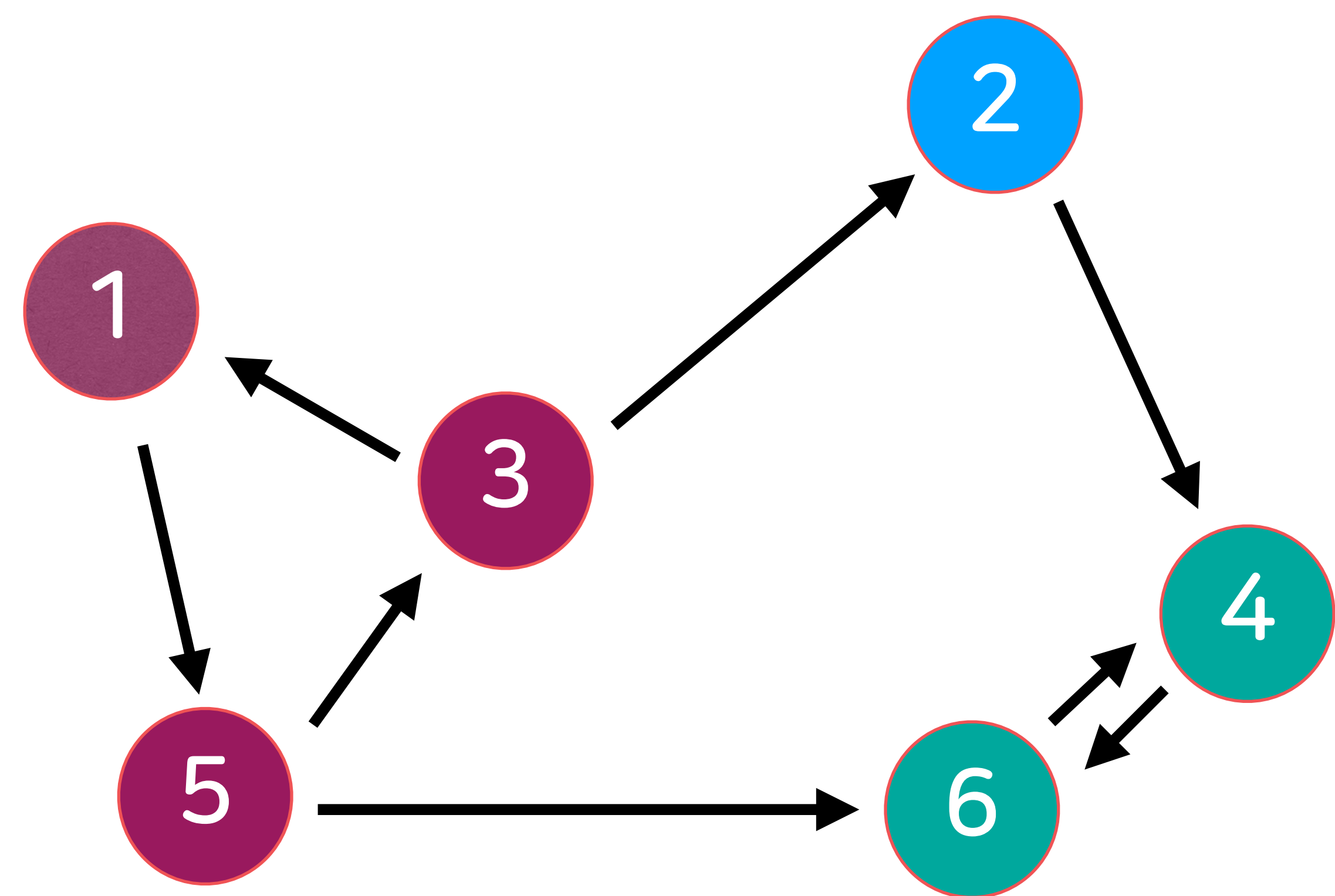
3. 역방향 그래프에 대해 DFS 진행 → 거쳐가는 모든 원소를 SCC로 묶기

어떻게 SCC를 찾지?

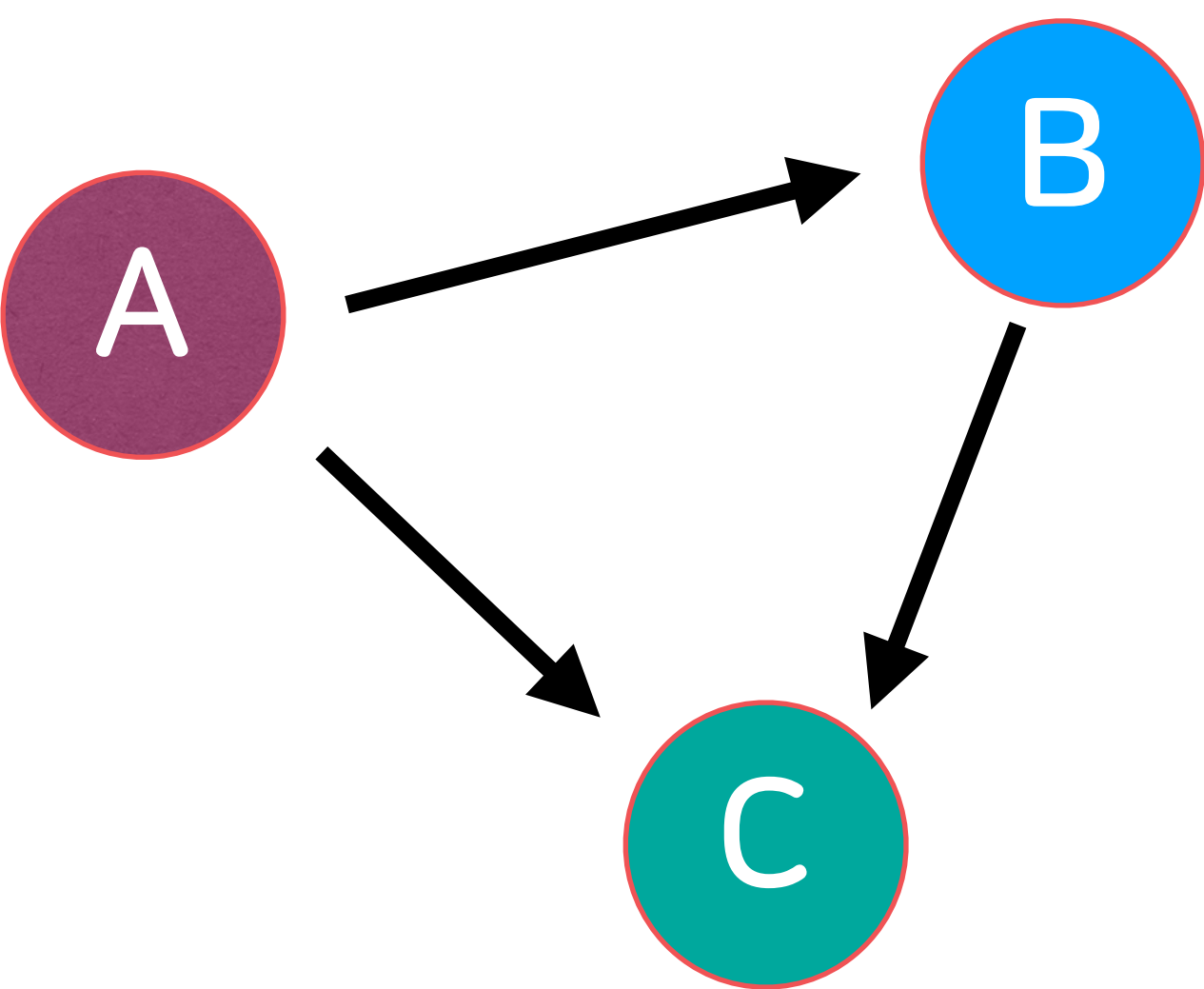
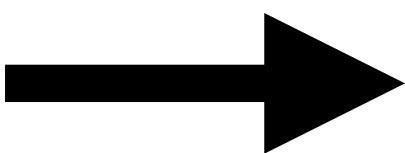


4. SCC 찾기 성공!

도시를 국가로!

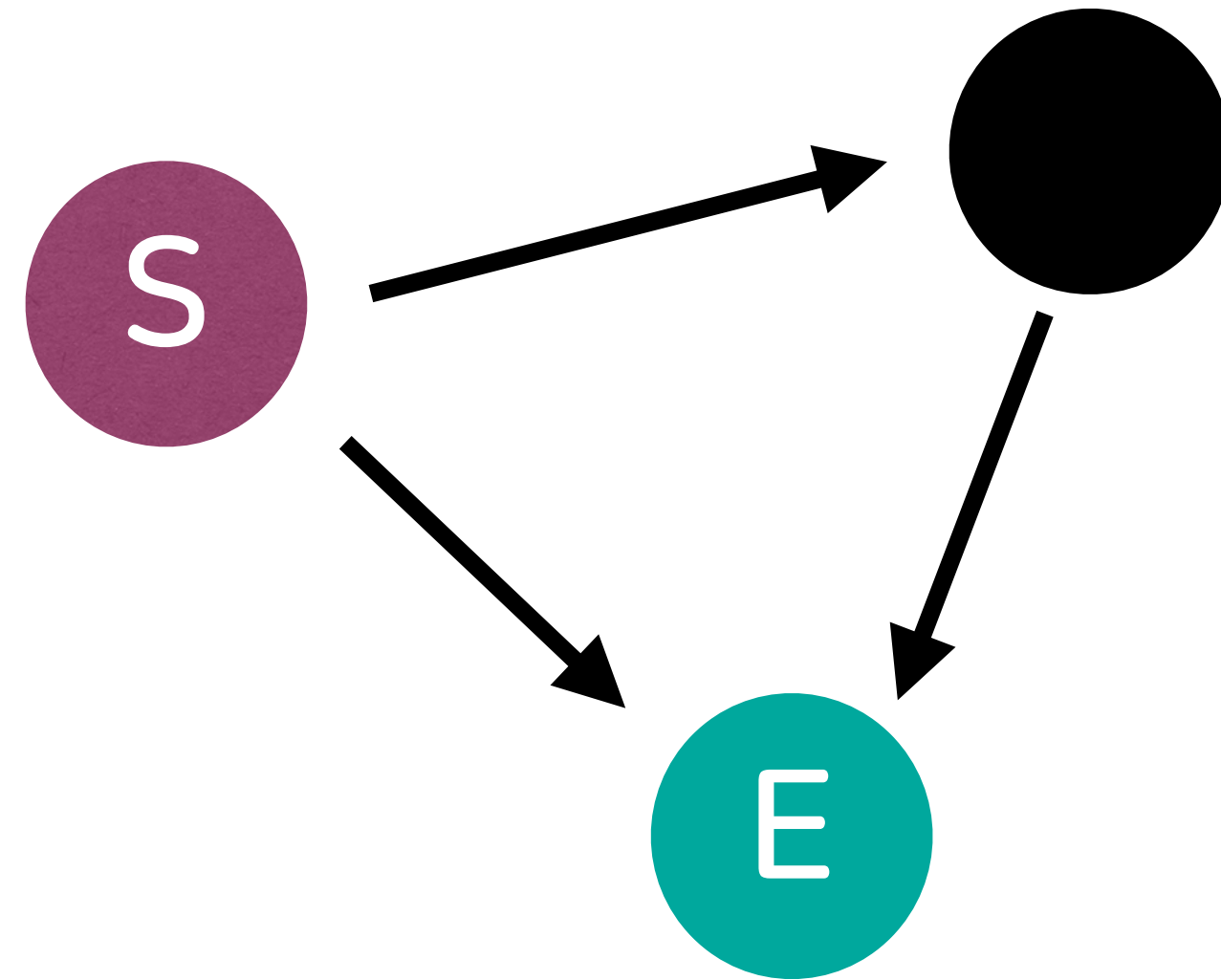


도시 그래프



국가 그래프

Finale



시작 도시가 속한 국가(**S**)와
도착 도시가 속한 국가(**E**)를
BFS로 최단 경로 탐색

함께 보면 좋은 문제들

BOJ 2150 : Strongly Connected Component

BOJ 4196 : 도미노

BOJ 3977 : 축구 전술

THANK YOU!



Hojun Lee
Dept. of Computer Science & Statistics
University of Seoul