

DFS 알고리즘

스택 자료구조와 재귀 함수

S.C.P

211202 이유경

Contents

1. 스택 자료구조

- 1-1. 스택 정의
- 1-2. 스택 동작 예시
- 1-3. 스택 구현 예제

2. 재귀 함수

- 2-1. 재귀 함수 정의
- 2-2. 재귀 함수 구현 예제

3. DFS 알고리즘

- 3-1. DFS 정의
- 3-2. DFS 동작 예시
- 3-3. DFS 구현 예제

1-1. 스택 정의

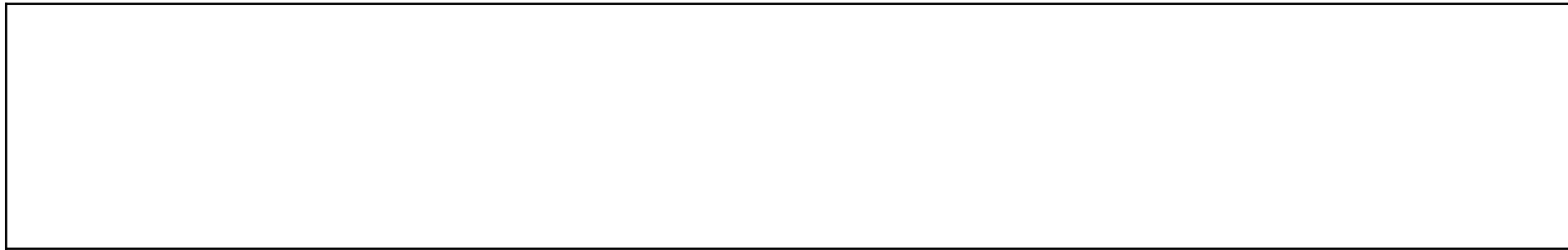
제한적으로 접근할 수 있는 나열 구조

후입선출 (*Last In First Out*)



1-2. 스택 동작 예시

삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



↑ 스택 바구니

1-2. 스택 동작 예시

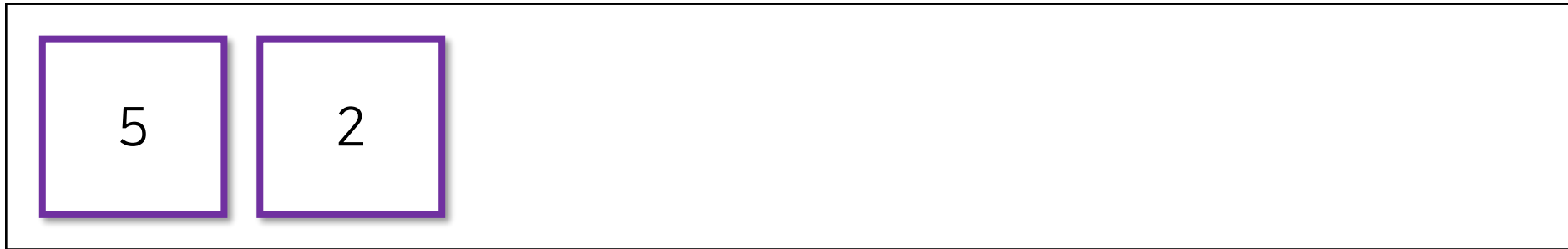
삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



5

1-2. 스택 동작 예시

삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



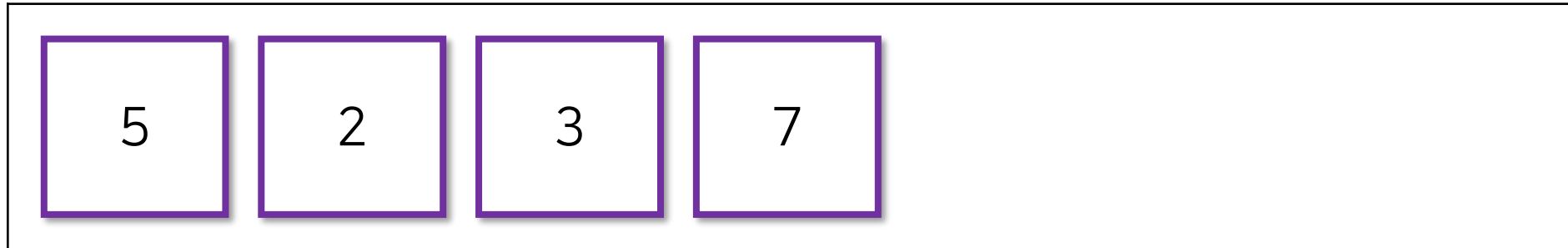
1-2. 스택 동작 예시

삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



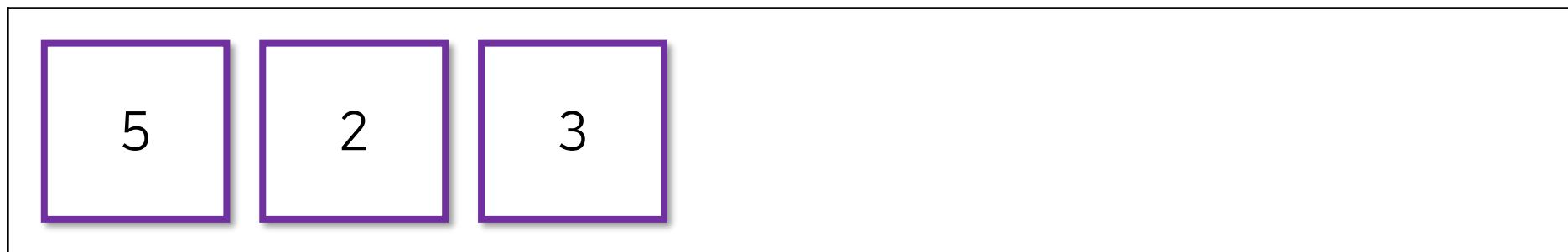
1-2. 스택 동작 예시

삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



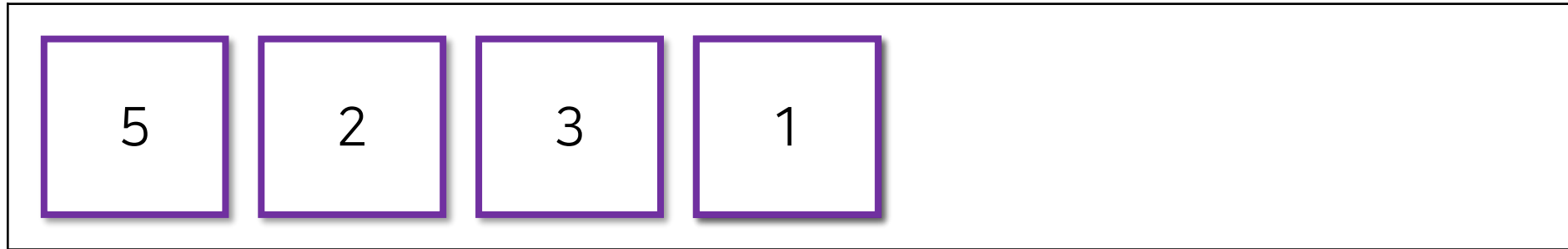
1-2. 스택 동작 예시

삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



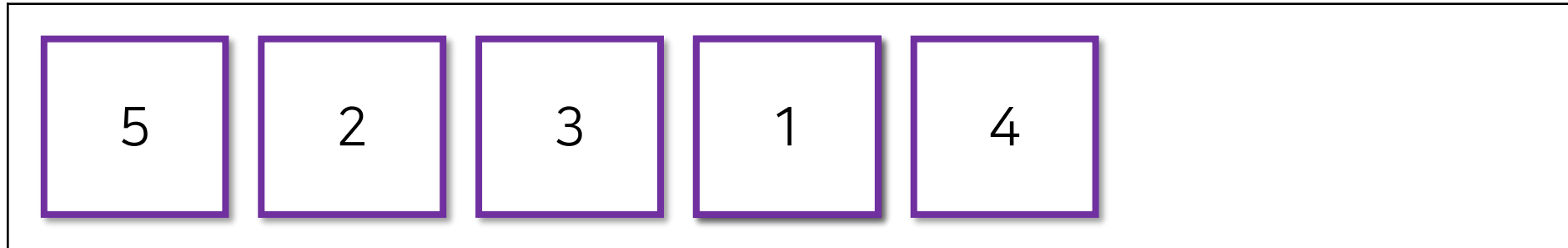
1-2. 스택 동작 예시

삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



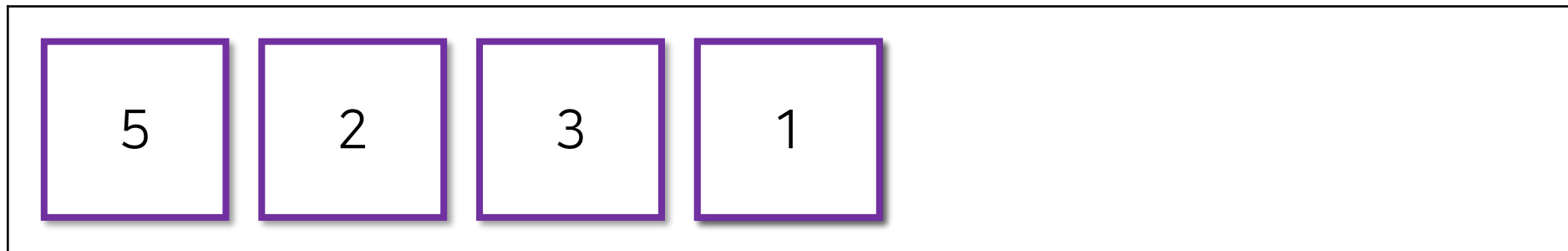
1-2. 스택 동작 예시

삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



1-2. 스택 동작 예시

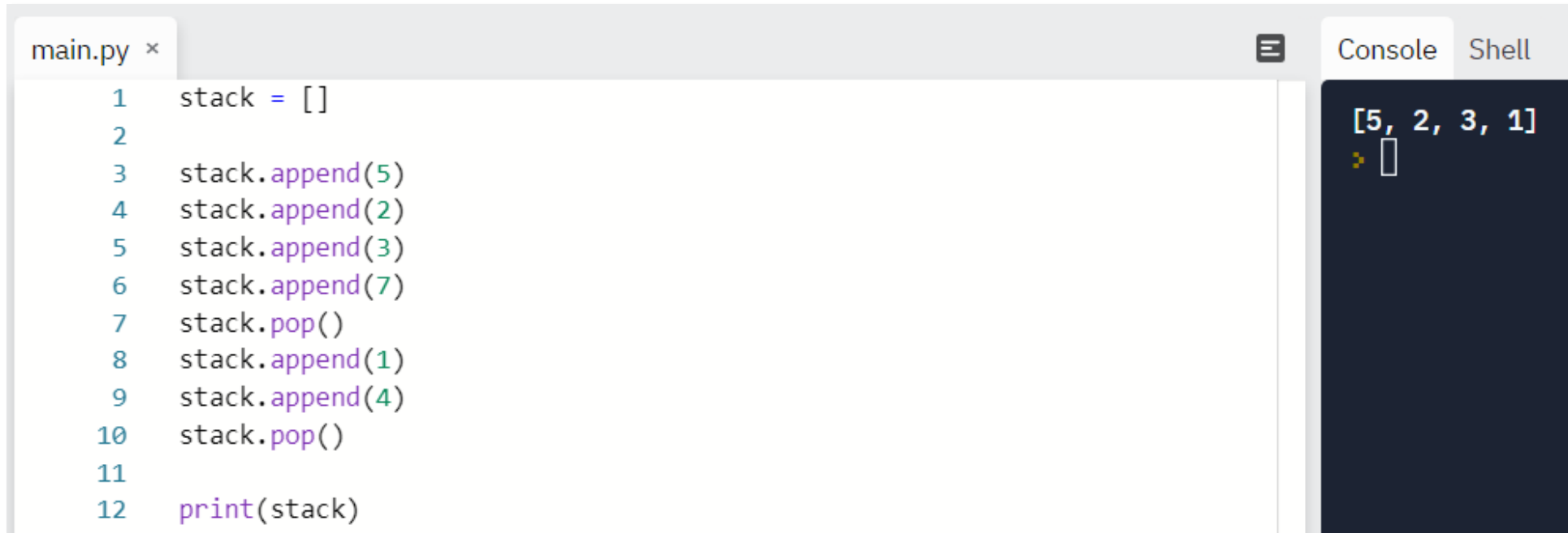
삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



1-3. 스택 구현 예제 (*Python*)

Python에서 리스트는 스택으로 구현되어 있다!

삽입(5) - 삽입(2) - 삽입(3) - 삽입(7) - 삭제() - 삽입(1) - 삽입(4) - 삭제()



The screenshot shows a code editor window titled 'main.py' with the following Python code:

```
1 stack = []
2
3 stack.append(5)
4 stack.append(2)
5 stack.append(3)
6 stack.append(7)
7 stack.pop()
8 stack.append(1)
9 stack.append(4)
10 stack.pop()
11
12 print(stack)
```

To the right of the code editor is a 'Console' tab. It displays the output of the program: `[5, 2, 3, 1]`. Below the output, there is a prompt character (a yellow dot) and an empty list `[]`, indicating the state of the stack after the final `pop()` operation.

2-1. 재귀 함수 정의



[트친소 중]코딩노예 덤실
@DevCode_in

어느 한 컴퓨터공학과 학생이 유명한 교수님을 찾아가 물었다.

"재귀함수가 뭔가요?"

"잘 들어보게. 옛날에 산 꼭대기에 현자가 있었어. 질문엔 모두 지혜롭게 대답해 주었지.

그런데 어느날, 그 선인에게 한 선비가 찾아와서 물었어.

"재귀함수가 뭔가요?"

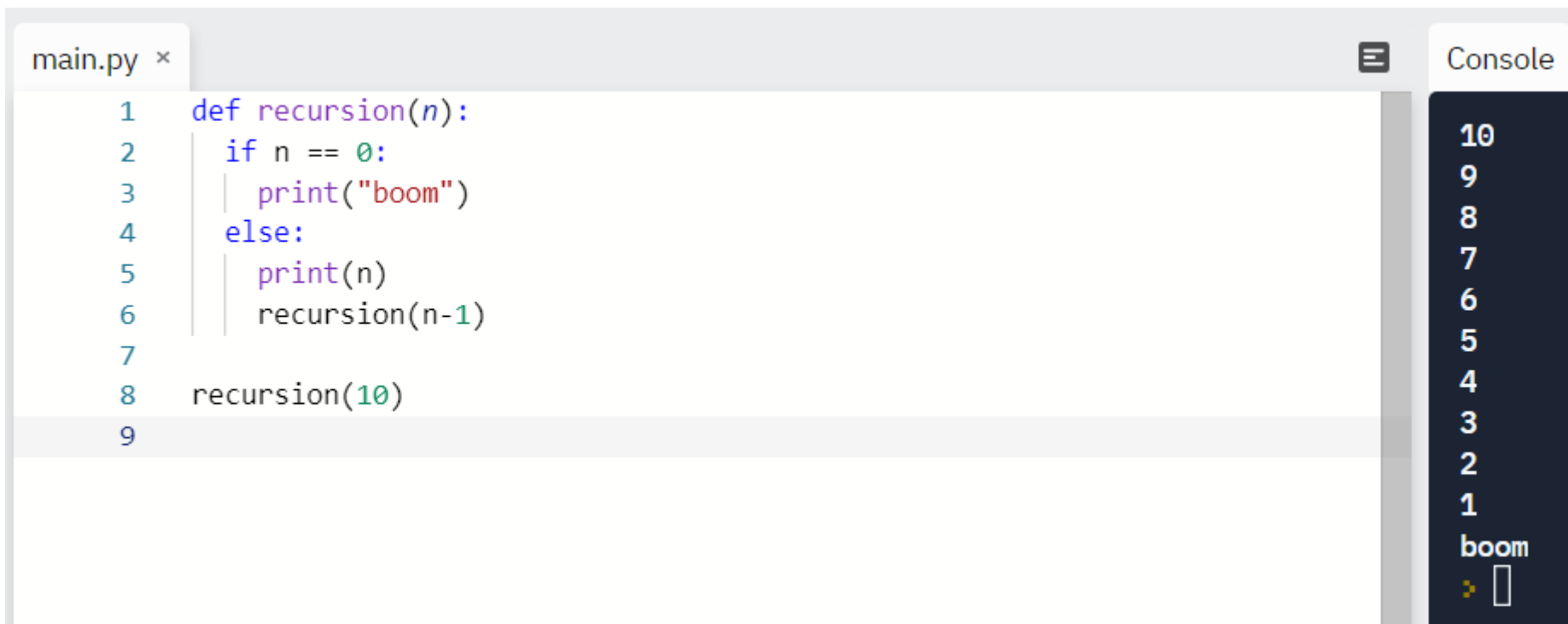
"잘 들어보게. 옛날에 산 꼭대기...

2017. 11. 10. 오전 12:45

자기 자신을 불러오는 함수

2-2. 재귀 함수 구현 예제 (Python)

카운트다운 재귀 함수



```
main.py x
1 def recursion(n):
2     if n == 0:
3         print("boom")
4     else:
5         print(n)
6         recursion(n-1)
7
8 recursion(10)
9
```

Console

10
9
8
7
6
5
4
3
2
1
boom
█

사실상 반복문과 다를 게 없다!

2-2. 재귀 함수 구현 예제 (유클리드 알고리즘)

유클리드 알고리즘: 두 개의 자연수에 대한 최대공약수 (GCD) 구하기

$GCD(A, B)$

	219	127
1	127	92
2	92	35
3	35	22
4	22	13
5	13	9
6	9	4
7	4	1

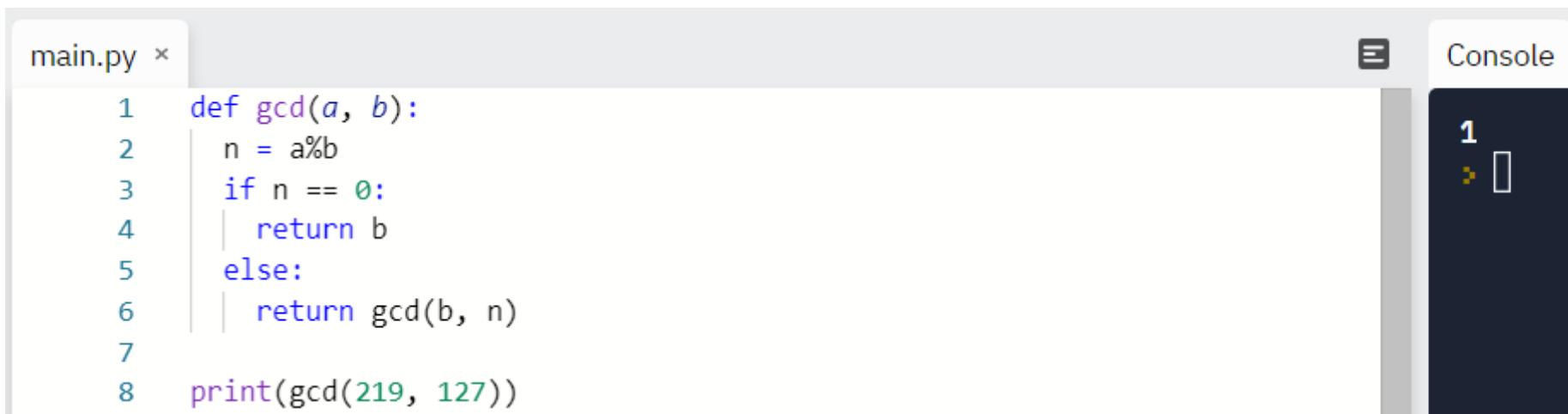
$$219 \% 127 = 92$$



2-2. 재귀 함수 구현 예제 (유클리드 알고리즘)

유클리드 알고리즘: 두 개의 자연수에 대한 최대공약수 (GCD) 구하기

$GCD(A, B)$



```
main.py x
1 def gcd(a, b):
2     n = a%b
3     if n == 0:
4         return b
5     else:
6         return gcd(b, n)
7
8 print(gcd(219, 127))
```

Console

```
1
> []
```

3-1. DFS 정의

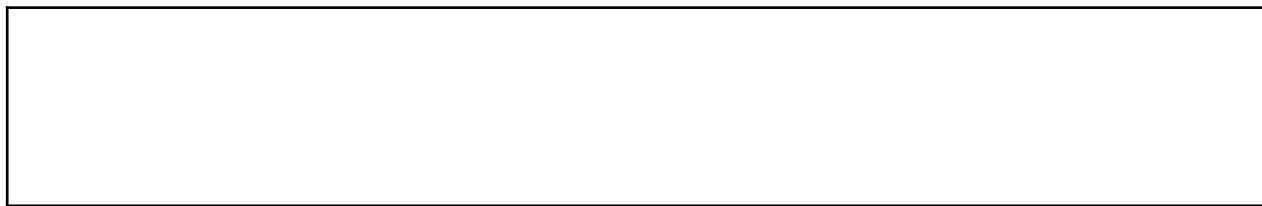
DFS(깊이 우선 탐색)

- 루트 노드에서 시작해서 다음 분기로 넘어가기 전에 해당 분기를 완벽하게 탐색하는 방법
- 스택 자료구조(혹은 재귀 함수)를 이용
- 모든 정점을 방문하는 단순하고 고전적인 방법

동작 과정

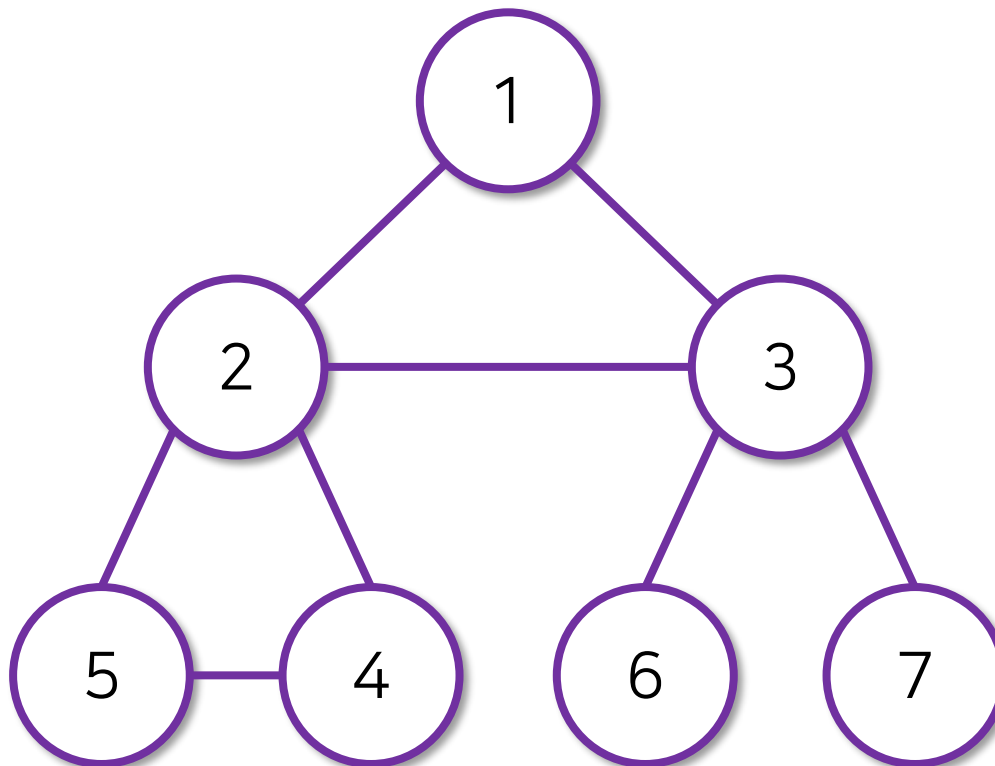
1. 스택의 최상단 노드 확인
2. 현재 정점과 인접한 간선들을 하나씩 검사
3. 아직 방문하지 않은 정점으로 향하는 간선이 있다면 그 간선을 무조건 방문
4. 더 이상 방문할 곳이 없다면, 마지막에 방문했던 간선을 따라 뒤로 돌아감

3-2. DFS 동작 예시

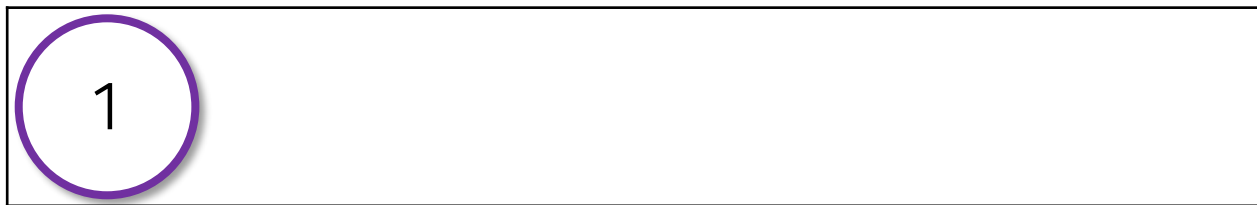


방문 기준

번호가 낮은 인접 노드부터

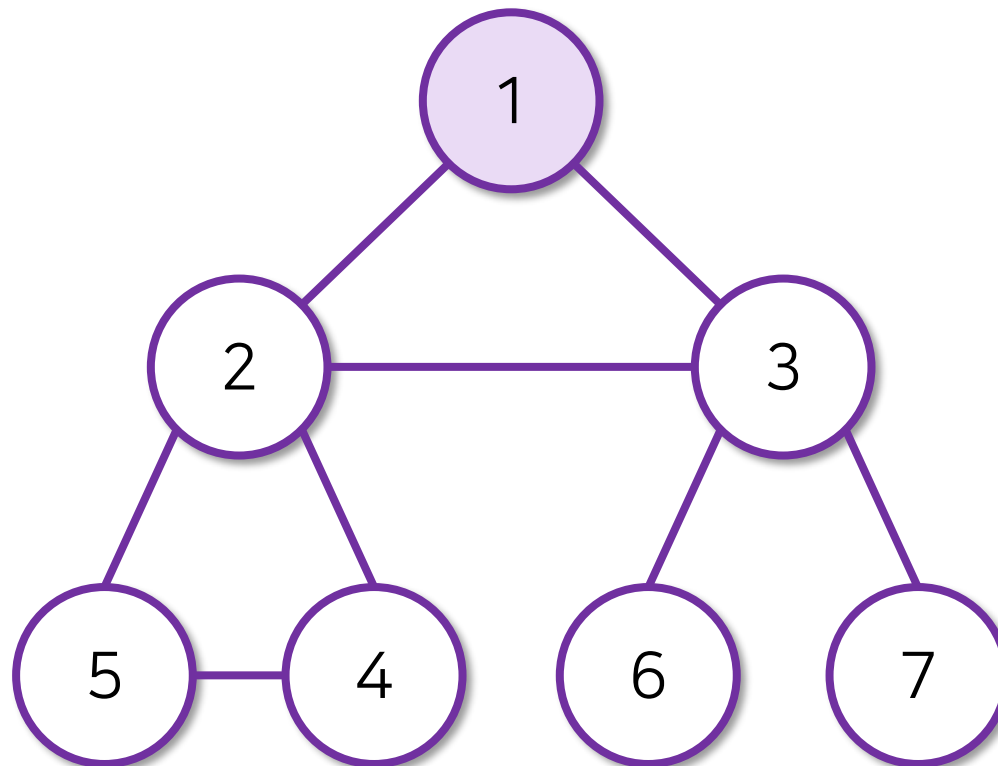


3-2. DFS 동작 예시

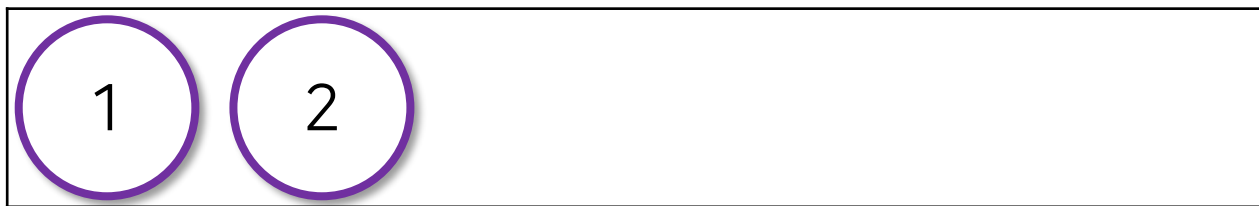


방문 기준

번호가 낮은 인접 노드부터

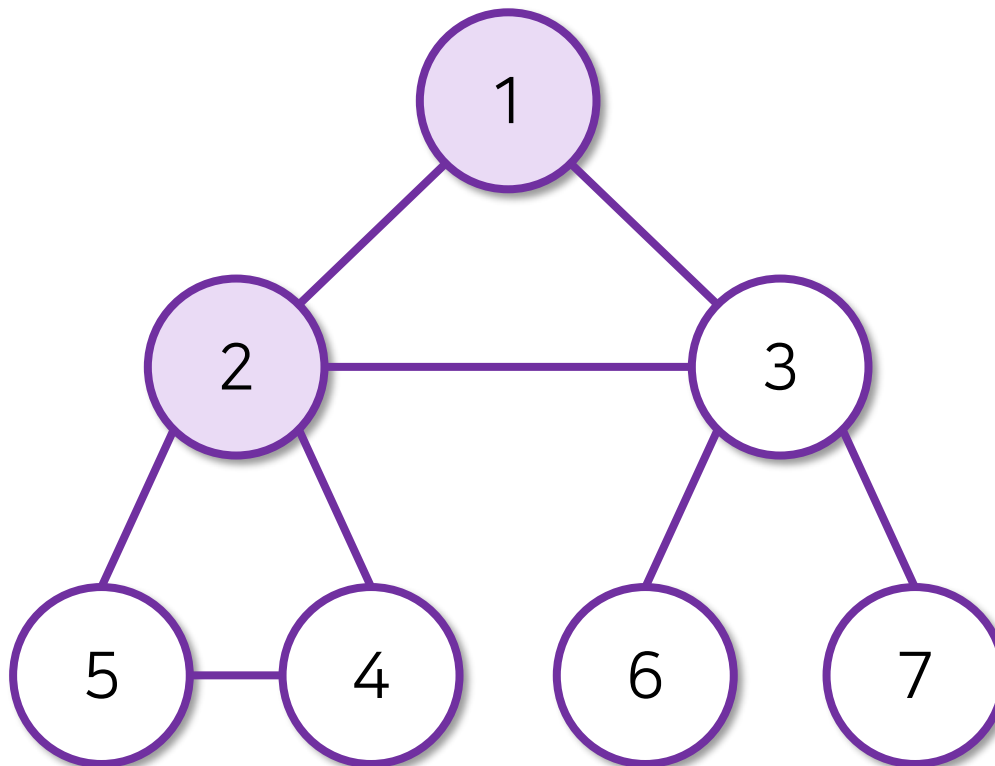


3-2. DFS 동작 예시

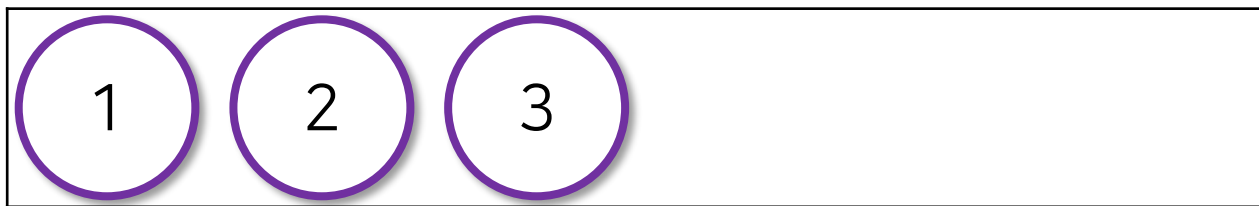


방문 기준

번호가 낮은 인접 노드부터

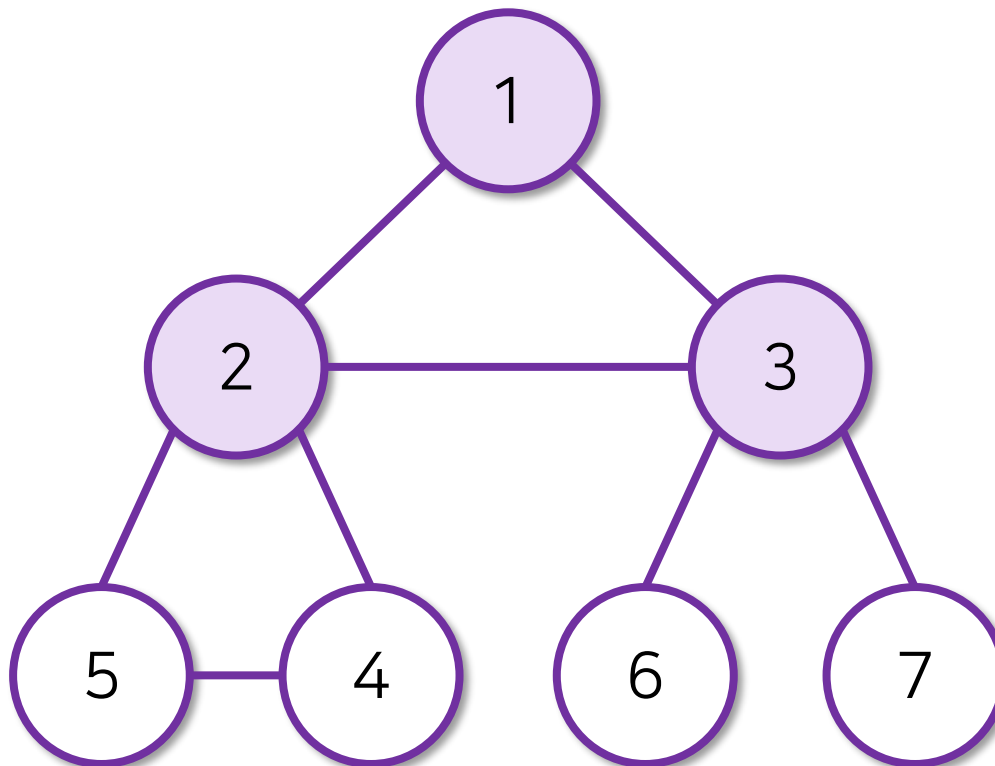


3-2. DFS 동작 예시

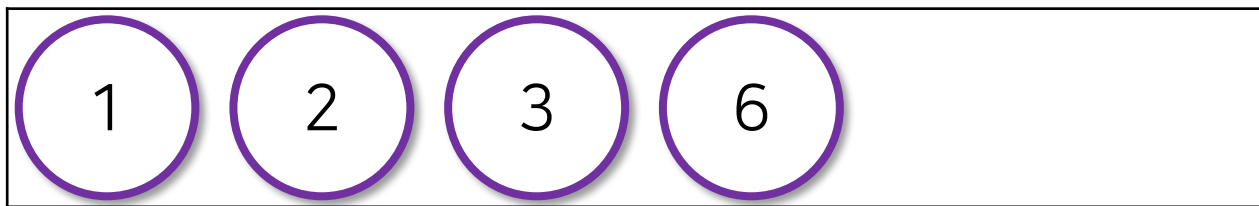


방문 기준

번호가 낮은 인접 노드부터

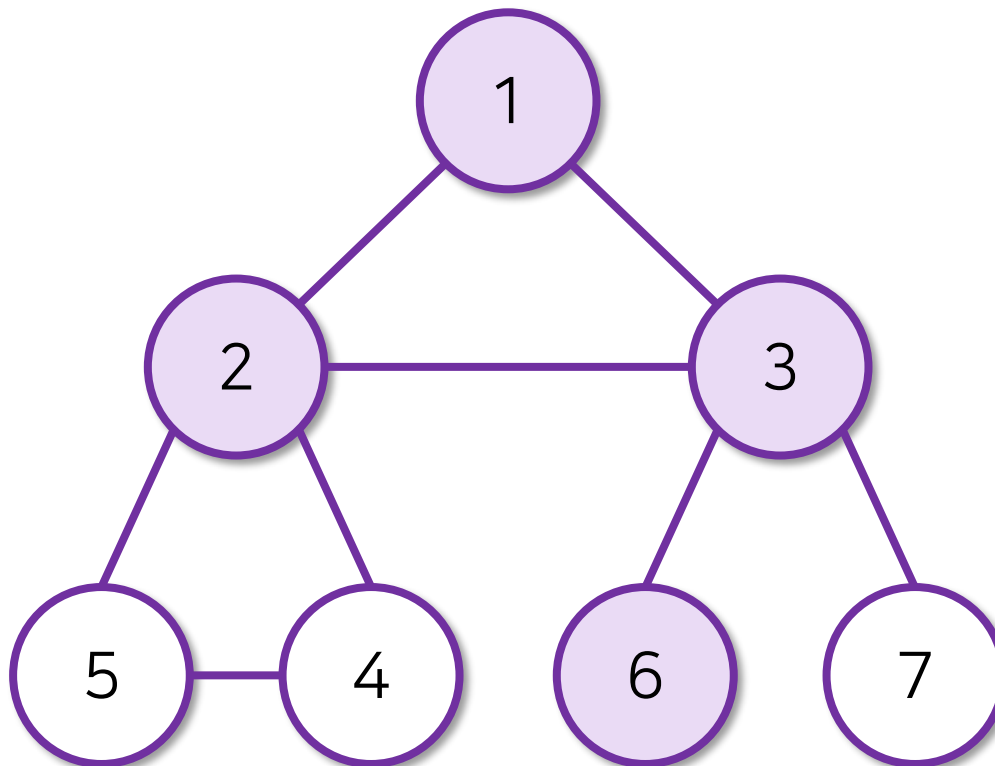


3-2. DFS 동작 예시

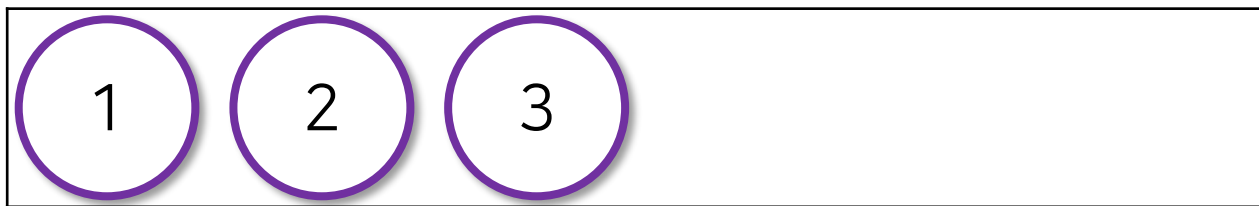


방문 기준

번호가 낮은 인접 노드부터

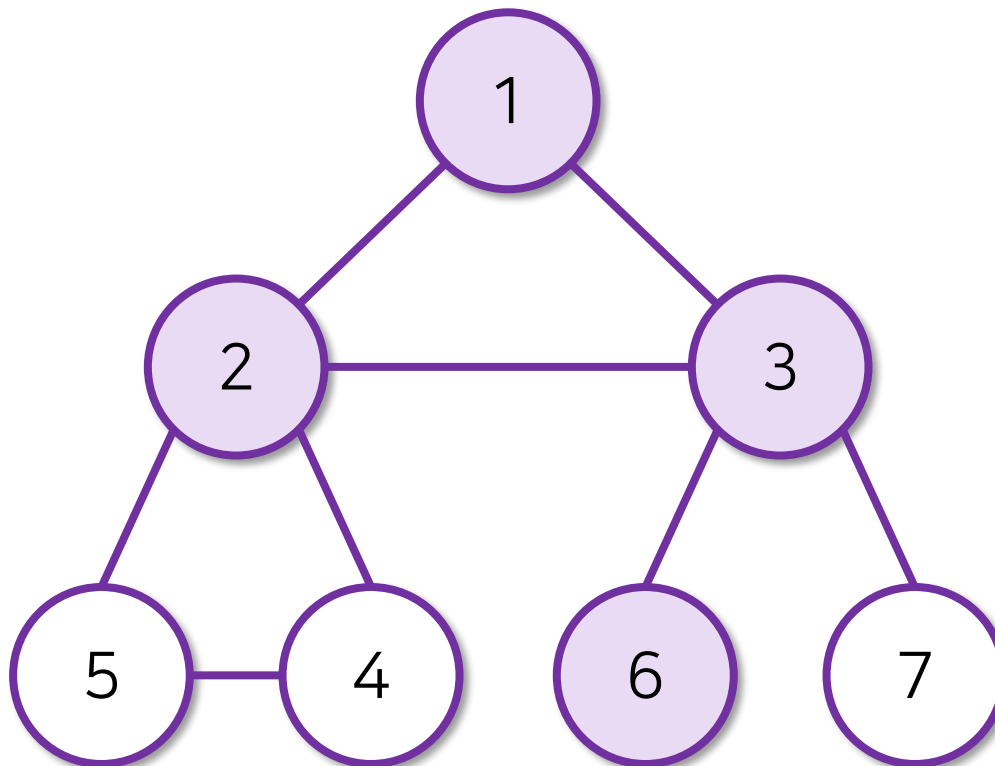


3-2. DFS 동작 예시

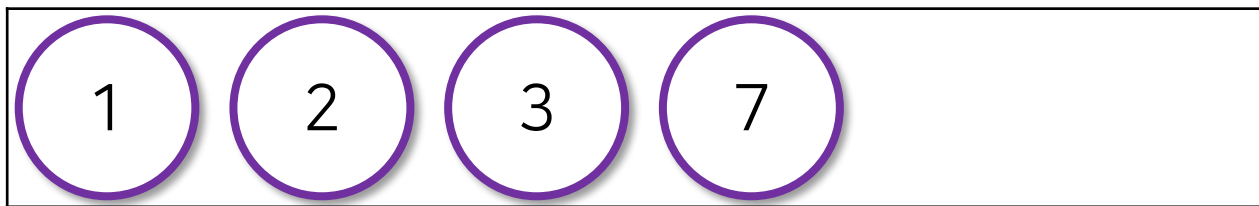


방문 기준

번호가 낮은 인접 노드부터

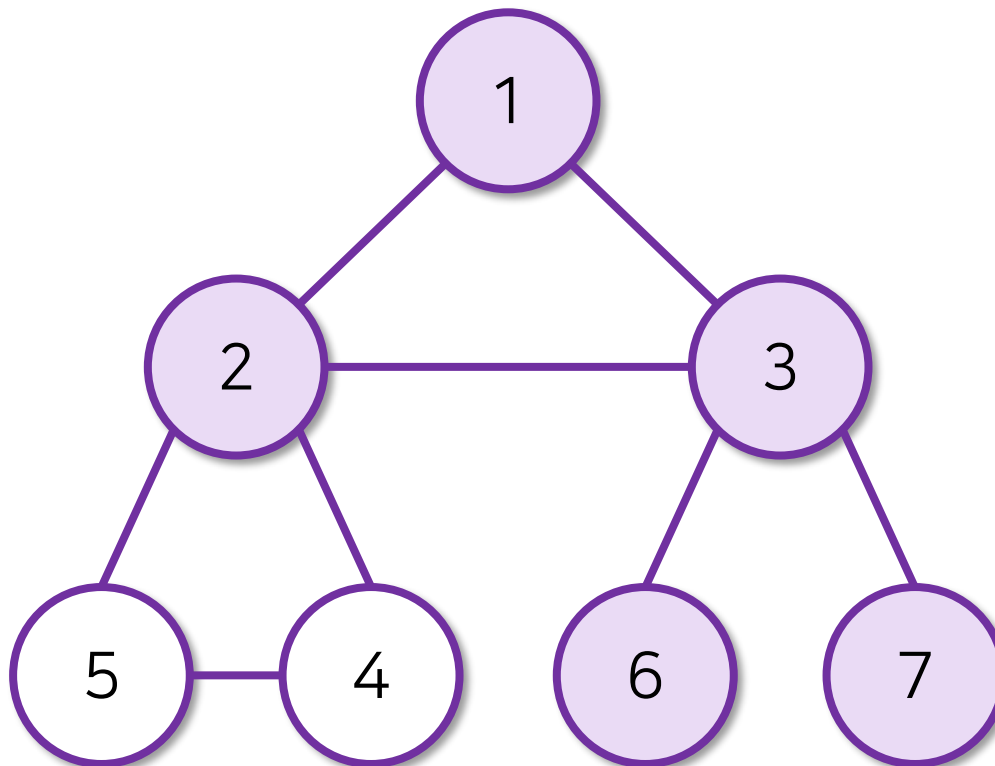


3-2. DFS 동작 예시

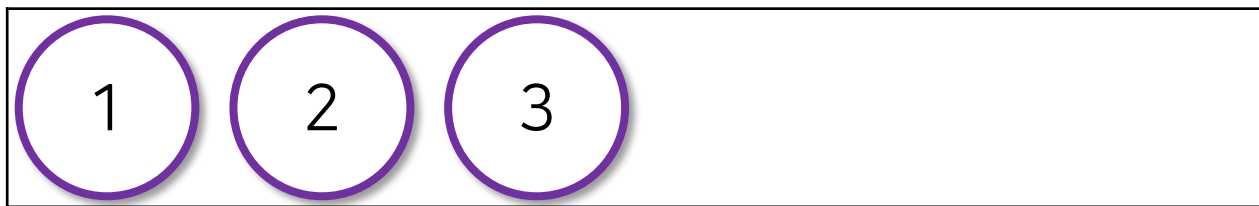


방문 기준

번호가 낮은 인접 노드부터

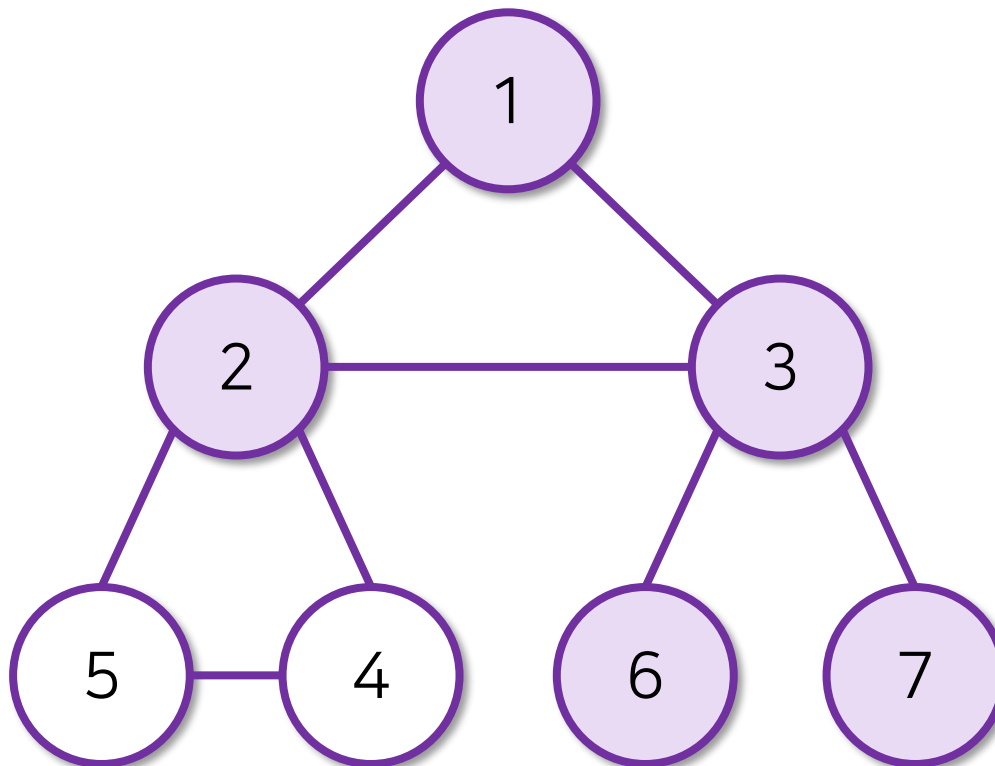


3-2. DFS 동작 예시

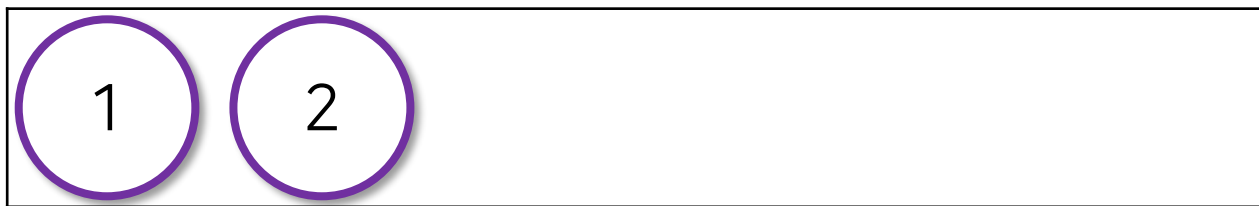


방문 기준

번호가 낮은 인접 노드부터

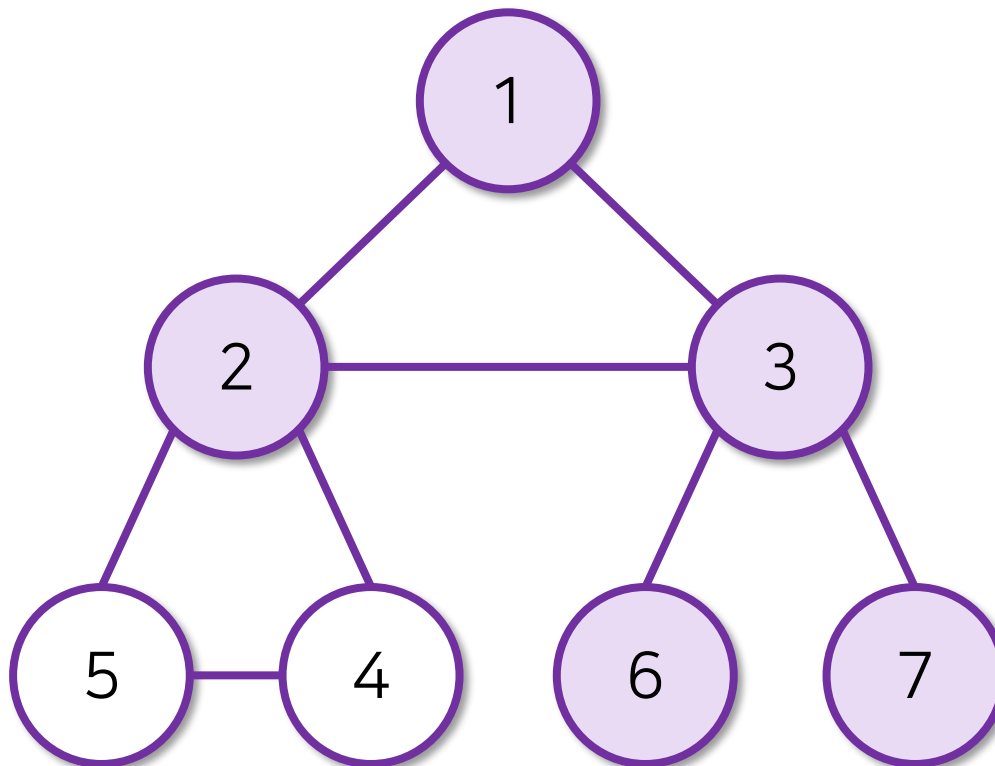


3-2. DFS 동작 예시



방문 기준

번호가 낮은 인접 노드부터

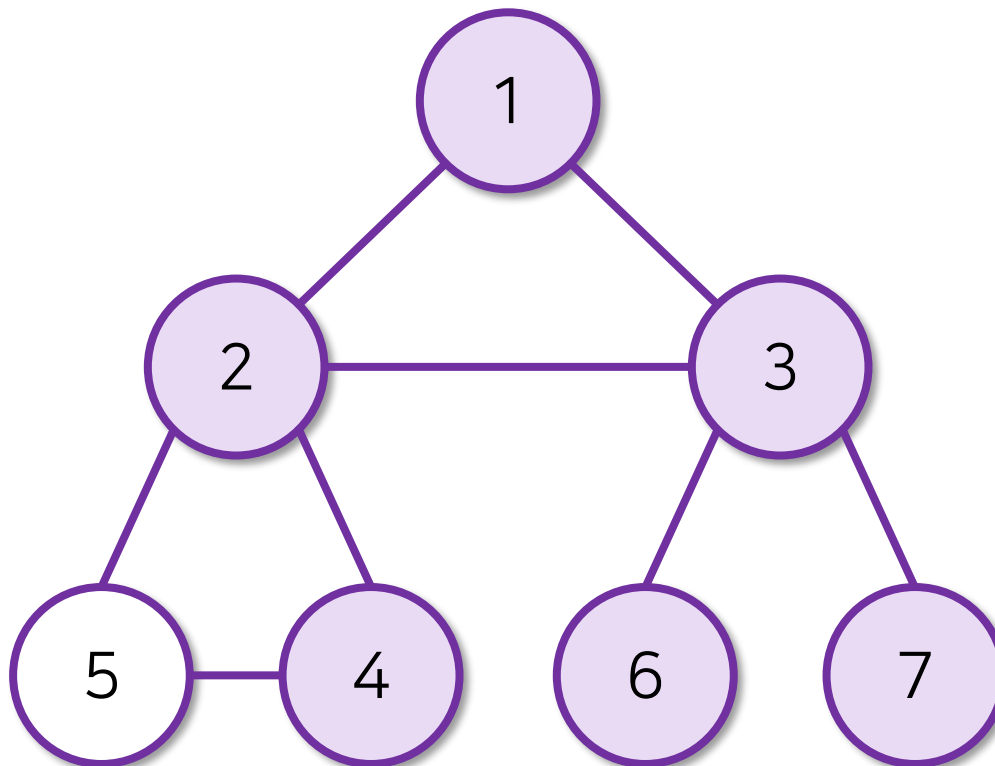


3-2. DFS 동작 예시

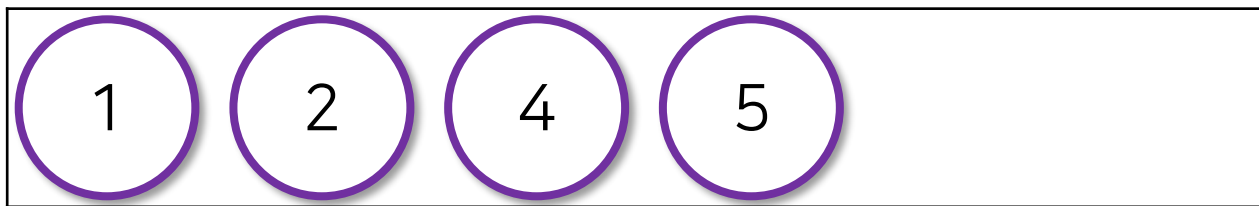


방문 기준

번호가 낮은 인접 노드부터

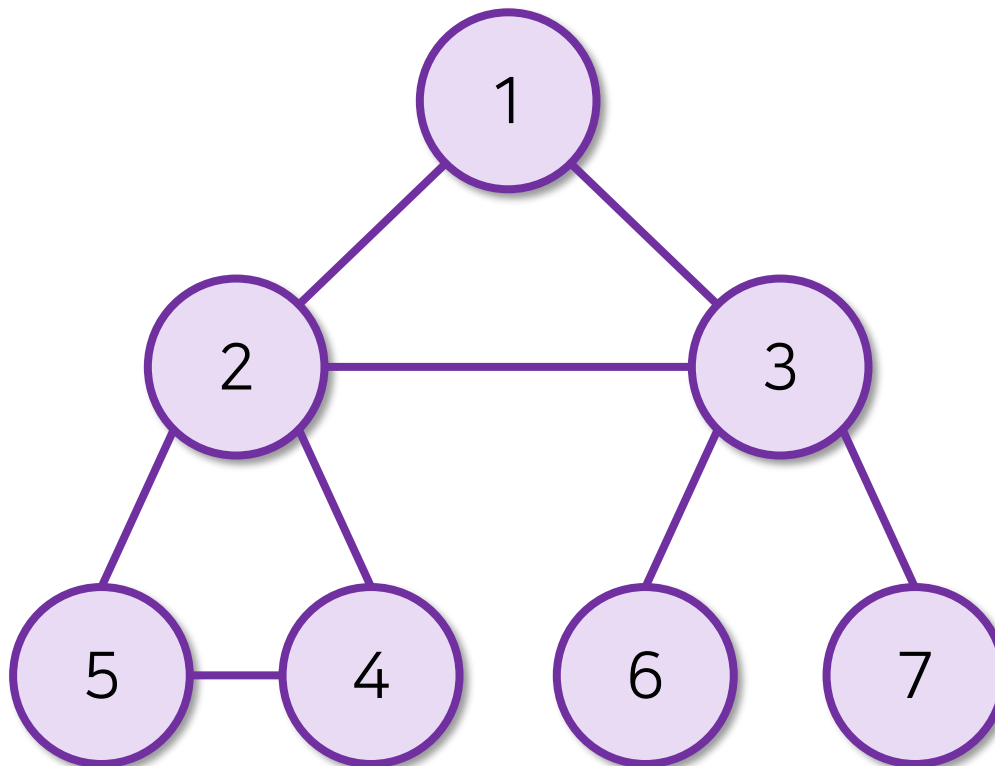


3-2. DFS 동작 예시



방문 기준

번호가 낮은 인접 노드부터

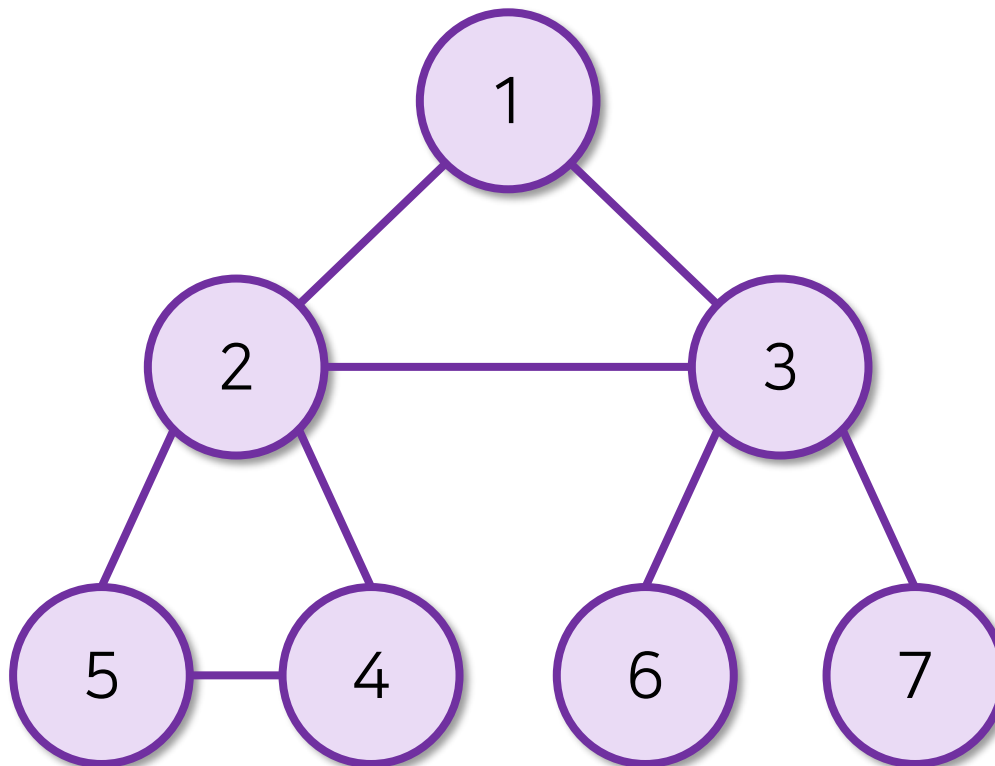


3-2. DFS 동작 예시

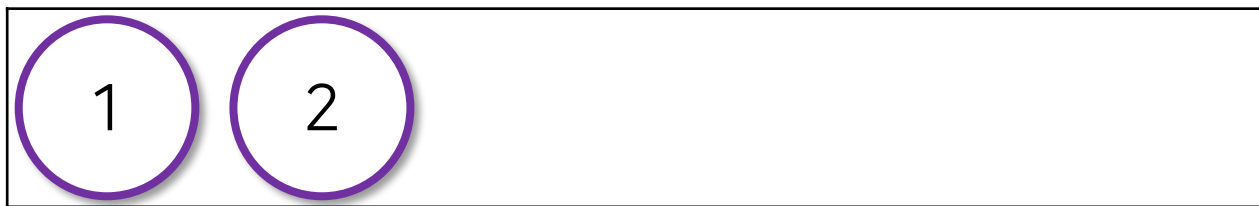


방문 기준

번호가 낮은 인접 노드부터

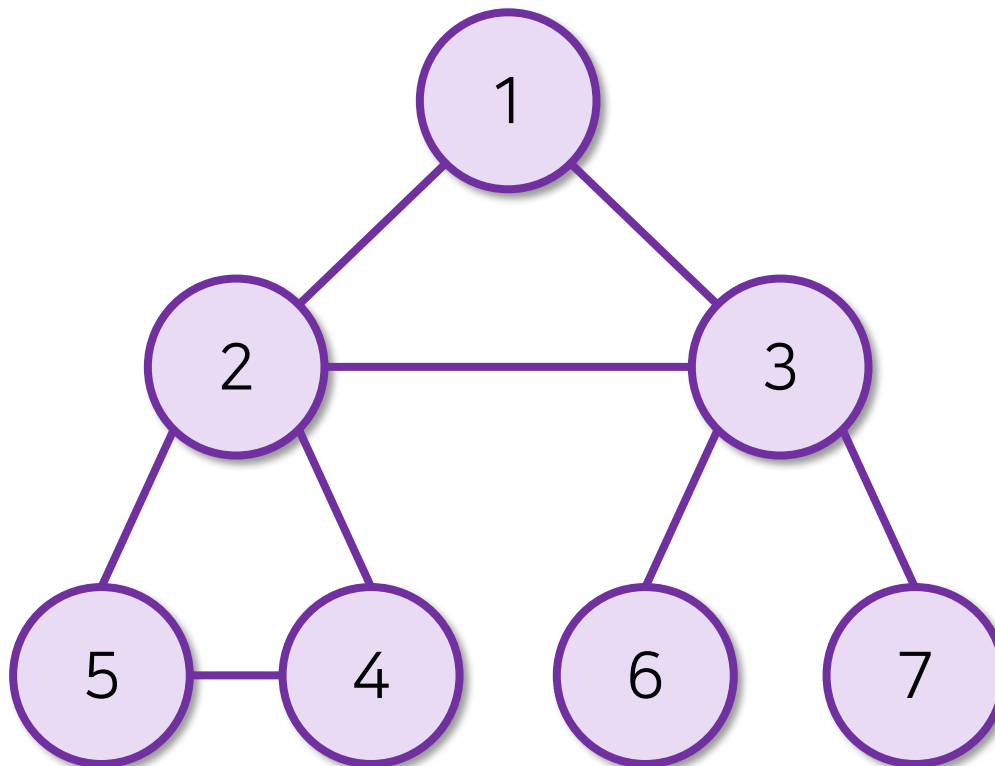


3-2. DFS 동작 예시

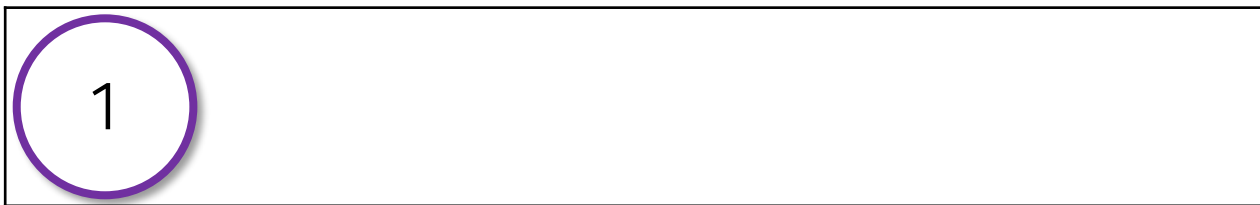


방문 기준

번호가 낮은 인접 노드부터

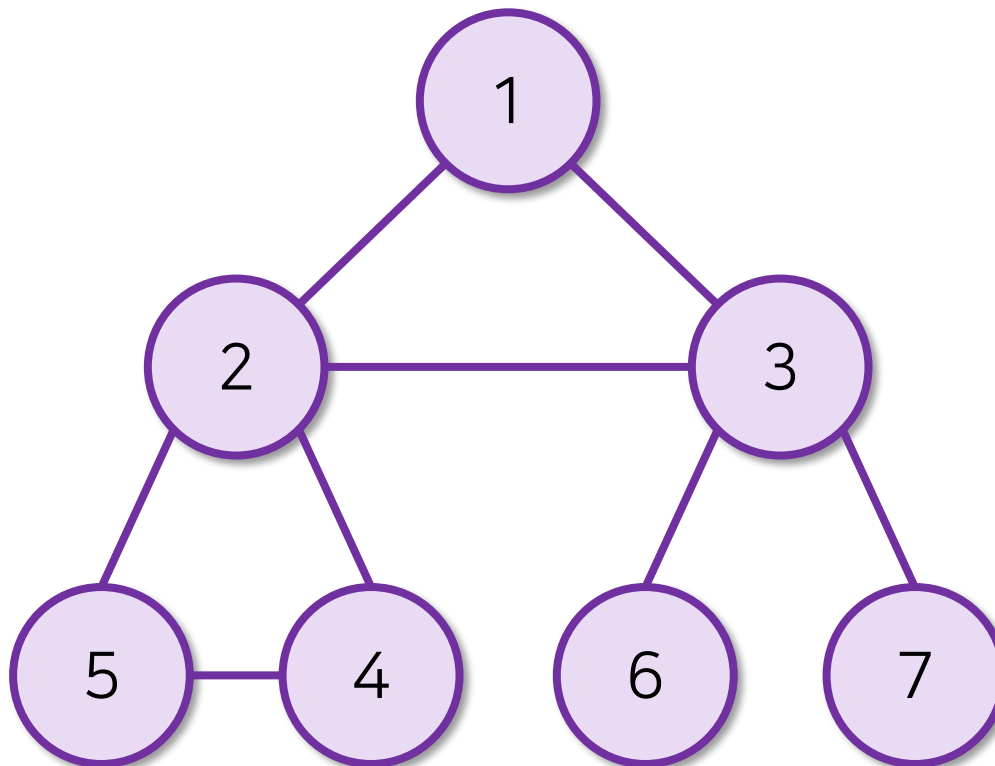


3-2. DFS 동작 예시

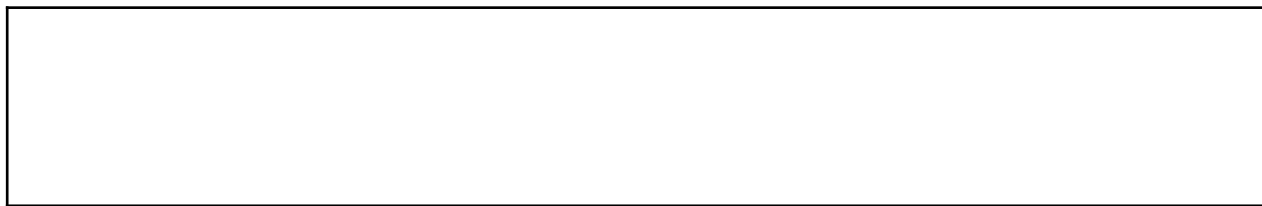


방문 기준

번호가 낮은 인접 노드부터

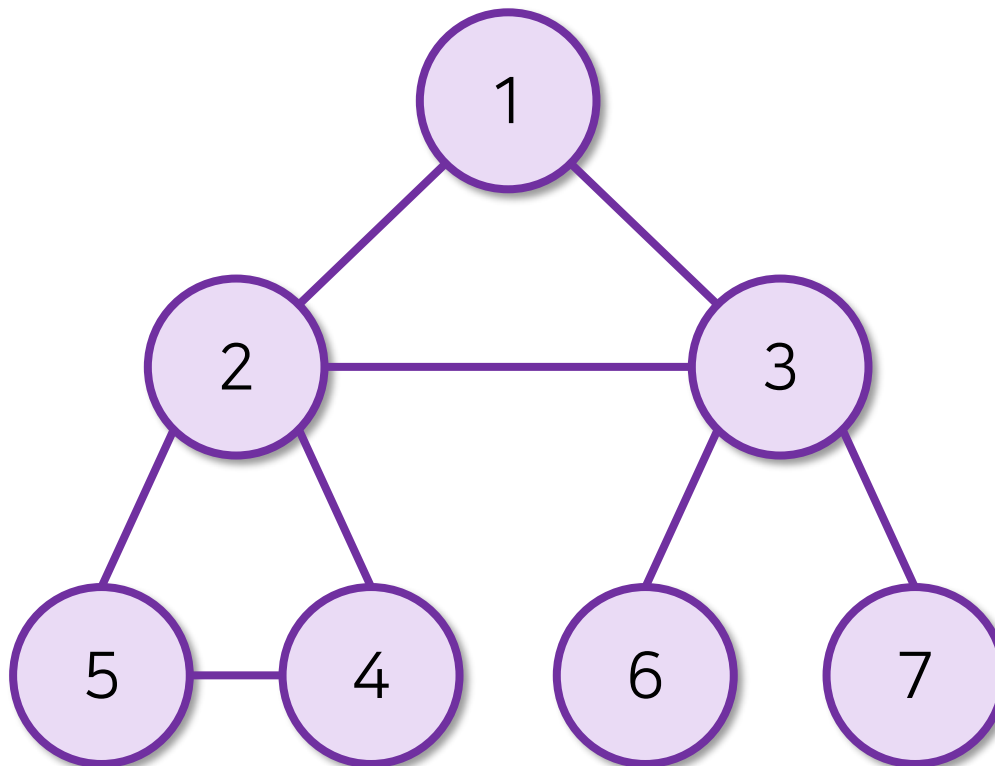


3-2. DFS 동작 예시



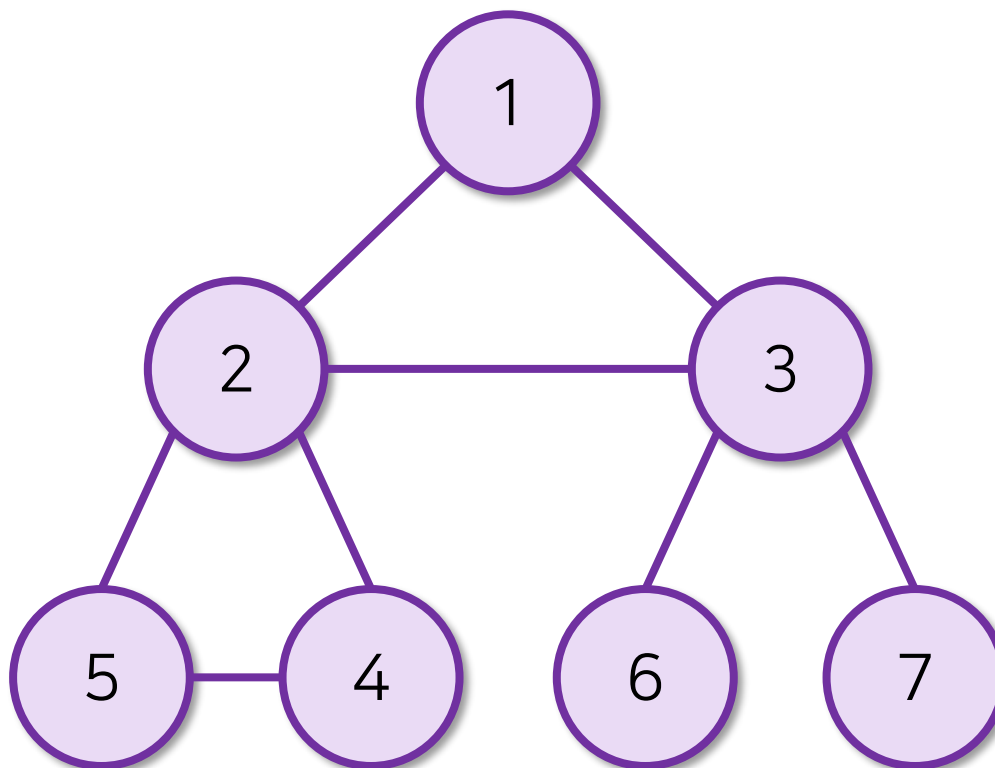
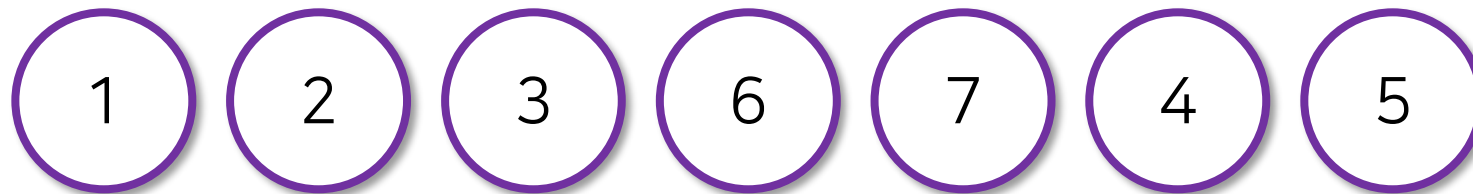
방문 기준

번호가 낮은 인접 노드부터

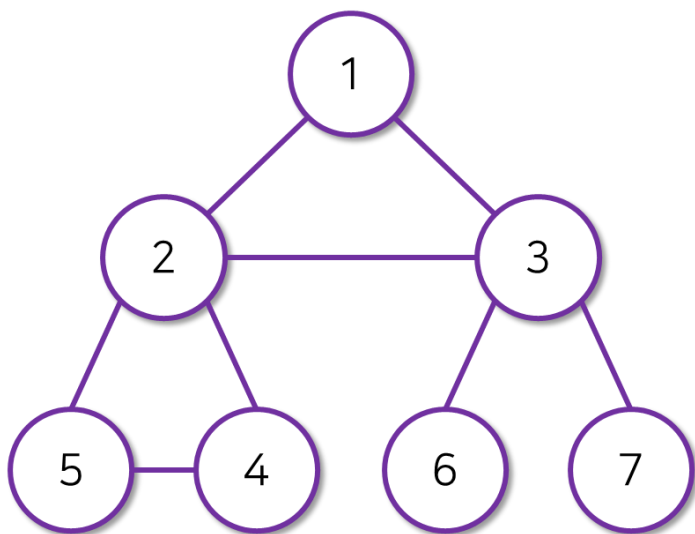


3-2. DFS 동작 예시

방문 경로



3-3. DFS 구현 예제 (Python)



```
1 def dfs(graph, v, visited):  
2     visited[v] = True  
3     print(v)  
4     for i in graph[v]:  
5         if not visited[i]:  
6             dfs(graph, i, visited)
```

main.py ×

```
1 #연결된 노드 2차원 리스트로 표현  
2 graph = [  
3     [],  
4     [2, 3],  
5     [1, 3, 4, 5],  
6     [1, 2, 6, 7],  
7     [2, 5],  
8     [2, 4],  
9     [3],  
10    [3]  
11 ]
```

```
12  
13 #방문한 노드 1차원 리스트로 표현  
14 visited = [False] * len(graph)
```

```
15  
16 def dfs(graph, v, visited):  
17     #현재 노드 방문 처리  
18     visited[v] = True  
19     print(v)  
20     #현재 노드와 연결된 다른 노드 재귀적 방문  
21     for i in graph[v]:  
22         if not visited[i]:  
23             dfs(graph, i, visited)  
24
```

```
25 #0번 노드가 없으니 1번 노드부터 탐색  
26 dfs(graph, 1, visited)
```

Console

```
1  
2  
3  
6  
7  
4  
5  
[]
```

THANK YOU

QUESTION
