**<5장 DFS/BFS>**

-탐색 🡺많은 양의 데이터중 원하는 데이터를 찾는 것.

-자료구조? 🡺데이터를 표현하고 관리하고 처리하기 위한 구조. 대표적으로 스택과 큐가 있다. Push와 POP으로 이뤄짐

-스택? 🡺박스 쌓기 선입 후출, 나중에 넣은 자료가 먼저 나온다. 리스트에서 append를 하면 삽입이고 pop을하면 출력이다.

-큐? 🡺대기열과 같다 먼저 들어온 놈이 먼저 나간다. Import deque from collections에서 deque.append, popleft등을 사용해서 출력 가능하다.

-재귀 함수?

Def recursive()

Print()

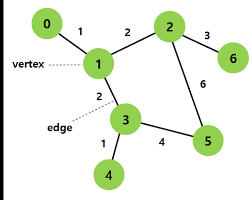
Recursive() 🡺이렇게 **함수에서 자기를 다시 호출하는 함수를 재귀 함수**라고 한다. 보통은 매개변수를 줘서 매개변수를 if 문으로 해서 걸러준다. (무한 루프 방지용)

<DFS> 🡪깊이 우선 탐색

-그래프에서 가장 깊은 부분을 우선적으로 탐색한다

-그래프? 🡺 그래프는 노드와 간선(Edge)로 이뤄진 구조이다. 그래프 탐색이란 따라서 하나의 노드를 시작으로 다수의 노드를 방문하는 것을 말한다.

노드는 도시 간선은 도로라 생각하면 이해하기 쉽다.

🡪두 노드가 간선으로 연결되어 있다면 두 노드는 **인접하다**!

-그래프의 표현 방식 2가지

***1)인접 행렬 2)인접 리스트***

말 그래도 그래프연결 관계를 행렬로 표현할지 리스트로 표현 할지에 따라 다르다. 두가지 다 이해를 하고 있어야 된다.

<인접 행렬 방식>

-**2차원 배열로 그래프의 연결 관계를 표현한다**. 파이썬은 행렬을 리스트로 구현 가능하긴 함

텍스트이(가) 표시된 사진

자동 생성된 설명

<인접 리스트 방식>

-각 노드의 연결 관계를 하나하나 append 시켜주는 방식 연결 리스트라는 것이 필요하지만 파이썬은 자동 제공(append로)

2차원 리스트를 초기화 시켜주고 연결 정도를 각 행에 넣어주면 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명

인접 행렬과 리스트의 가장 큰 차이? 🡺인접 행렬은 한번에 모든 데이터를 한번에 초기화? 리스트는 일일히 정보를 모두 입력해야 된다. 즉 **정보를 빠르게 얻는 쪽**은 인접 행렬이다. (그냥 graph[0][1] 이렇게 확인만 하면되기 때문이다.)

하지만 **메모리 사용 효율 측면**에서는 인접 리스트가 더욱 효율적이다. 왜냐면 인접 행렬은 모든 연결 관계를 저장하고 인접리스트는 연결된 정보만을 저장하기 때문(INF이런게 없음)

<DFS의 작동 방식> 🡺***stack 방식***으로 처리한다!

1)탐색 시작 노드를 스택에 삽입하고 방문 처리

2)스택의 최상단에서 노드에 **방문하지 않은 인접 노드가 있다면** 인접 노드를 스택에 넣고 방문처리! 만약 방문하지 않은 인접 노드가 없다면 스택에서 최 상단 노드를 pop!

3) 1-2를 계속 반복해 수행이 안될 때까지 반복한다.

여기서의 방문처리? 🡺 **스택에 삽입 되 처리된 노드는 다시 삽입되지 않도록 체크해 두는 것**이다. 따라서 각 노드는 한 번만 처리 과정을 겪도록 한다. Why? 🡺인접 노드가 더 처리되기 전까지는 stack에 있기 때문

<실제 구현 예제>🡺 재귀 함수를 사용해 간결하게 구현 가능

텍스트이(가) 표시된 사진

자동 생성된 설명

재귀 함수가 핵심이다. WHY? 🡺 처음 시작점을 주어주면 재귀함수로 모든 구역을 확인할 수 있기 때문이다.

<BFS>-너비 우선 탐색(Breadth First Search)

-가장 **인접한 노드부터 탐색하는 알고리즘**이다. DFS는 가장 멀리 있는 노드부터 탐색하는 방식이라면 BFS는 주변부터 확실하게 탐색해 가면서 영역을 확장하는 개념임.

-DFS와 다르게 스택이 아닌 **Queue방식**을 활용한다. 따라서 먼저 들어온 노드가 먼저 pop된다. 추가적으로 BFS는 동작이 더 빠르다!

<실제 코드>

텍스트이(가) 표시된 사진

자동 생성된 설명

알고리즘 순서

1)시작점을 queue=deque([start])로 queue를 초기화 해주는데 사용 이러면 queue는 [start]의 상태가 된다.

2)큐가 비어있는지 확인

3)큐의 맨 앞의 인자 값을 pop 시키고 print 해준다?

4)해당 인자와 인접한 모든 노드들을 for문으로 받아온다. 이때 방문 처리도 해준다.

5)2번부터 4번까지 계속 반복

**<정리>**

-탐색을 위해서는 그래프 구조와 인접 행렬, 인접 리스트 ,재귀 함수, 스택, 큐에 대한 개념이 필요로 한다.

-DFS는 스택과 재귀함수를 활용, BFS는 Queue를 활용한다. BFS가 조금 더 수행 시간이 짧다!

-DFS는 인접한 노드 하나를 계속 따라가는 방식(인접이 끝날 때 까지) BFS는 인접한 노드를 전부 받아오고 하나씩 꺼내면서 해당 노드와 인접한 노드를 위로 쌓는 방식이다.