그래프 탐색- DFS,BFS

practice 자료구조 · 2024. 10. 14. 11:49

그래프 탐색 알고리즘 중 **깊이 우선 탐색(DFS: Depth-First Search)**와 **너비 우선 탐색(BFS: Breadth-First Search)**은 그래프나 트리의 모든 노드를 방문하기 위한 대표적인 알고리즘입니다. 두 알고리즘은 서로 다른 방식으로 노드를 방문하며, 특정 상황에 따라 적합한 탐색 방법이 달라집니다. 각 알고리즘의 특징과 동작 방식을 설명하겠습니다.

1. 깊이 우선 탐색 (DFS: Depth-First Search)

개념
DFS는 그래프에서 **최대한 깊이**로 탐색한 후, 더 이상 갈 곳이 없으면 **다시 되돌아오는(backtracking)** 방식으로 동작합니다. 즉, 한 노드에서 출발하여 다음 노드를 탐색할 때, 더 깊이 들어갈 수 있을 때까지 계속 탐색하다가 막히면 직전 노드로 돌아가 다른 경로를 탐색합니다.

동작 방식
1. 시작 노드를 스택에 넣습니다.
2. 스택에서 노드를 꺼내 해당 노드를 방문하고, 아직 방문하지 않은 인접한 노드를 스택에 넣습니다.
3. 이 과정을 반복하며 더 이상 방문할 노드가 없을 때까지 탐색합니다.

주요 특징
- **스택(Stack)** 또는 **재귀(Recursion)**를 이용해 구현할 수 있습니다.
- DFS는 **깊은 경로**를 우선 탐색하므로, 주로 **경로 탐색**에 유용합니다.
- **재귀 호출**을 통해 간단하게 구현할 수 있지만, 재귀 깊이에 따라 스택 오버플로우 위험이 있습니다.

시간 복잡도

- 그래프의 모든 노드를 한 번씩 방문하므로 시간 복잡도는 O(V + E)입니다. 여기서 V는 노드(정점)의 개수, E는 간선의 개수를 의미합니다.

DFS의 활용 예시
- **미로 탐색**: 한 경로를 끝까지 탐색한 후, 막히면 다른 경로로 이동하는 방식.
- **사이클 검출**: 그래프에 사이클이 존재하는지 여부를 탐색할 때 유용.

```
def dfs(graph, start, visited=set()):
    visited.add(start)
    print(start, end=' ')

    for neighbor in graph[start]:
        if neighbor not in visited:
            dfs(graph, neighbor, visited)
```

2. 너비 우선 탐색 (BFS: Breadth-First Search)

개념
BFS는 **가장 가까운 노드**부터 차례대로 탐색하는 방식입니다. 즉, 탐색을 할 때 현재 노드와 **인접한 모든 노드를 먼저 탐색**한 후, 그 다음으로 인접한 노드들의 이웃을 탐색합니다. 한 단계씩 탐색 영역을 넓혀가는 방식이라고 할 수 있습니다.

동작 방식
1. 시작 노드를 큐에 넣습니다.
2. 큐에서 노드를 꺼내 방문하고, 인접한 노드들을 모두 큐에 넣습니다.
3. 큐가 빌 때까지 이 과정을 반복하며 탐색을 진행합니다.

주요 특징
- **큐(Queue)**를 사용하여 구현합니다.
- BFS는 **가장 가까운 경로**를 우선 탐색하므로, **최단 경로 탐색**에 유용합니다.
- 재귀 호출 없이 **반복문**을 통해 구현됩니다.

시간 복잡도
- BFS 역시 그래프의 모든 노드를 한 번씩 방문하므로, 시간 복잡도는 O(V + E)입니다.

BFS의 활용 예시
- **최단 경로 탐색**: 가중치가 없는 그래프에서 두 노드 사이의 최단 경로를 찾을 때.
- **사회적 네트워크 분석**: 연결된 관계망에서 거리가 가까운 친구나 지인들을 먼저 탐색하는 방식.

```
from collections import deque

def bfs(graph, start):
    visited = set()
    queue = deque([start])
    visited.add(start)

    while queue:
        node = queue.popleft()
        print(node, end=' ')

        for neighbor in graph[node]:
            if neighbor not in visited:
                queue.append(neighbor)
                visited.add(neighbor)
```

DFS vs BFS 비교

특징	DFS	BFS
탐색 방식	깊이 우선 (먼저 깊이 탐색)	너비 우선 (먼저 넓이 탐색)
자료 구조	스택(Stack) or 재귀(Recursion)	큐(Queue)
적용 상황	경로 탐색, 사이클 검출	최단 경로 탐색
시간 복잡도	O(V + E)	O(V + E)
공간 복잡도	O(V) (최악의 경우)	O(V)
주요 특징	경로의 깊이를 먼저 탐색, 막히면 백트래킹	최단 경로 보장, 가까운 노드부터 탐색

DFS는 주로 **탐색의 경로**가 중요한 경우에 사용되고, BFS는 **최단 경로 탐색**과 같이 **가장 가까운 노드**를 우선으로 탐색하는 경우에 적합합니다.

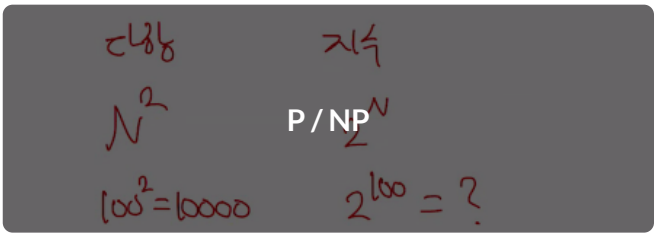
---

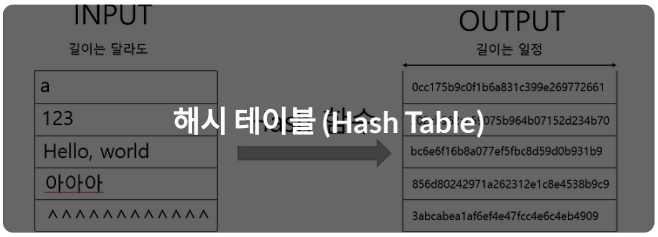
https://youtu.be/y3zZDFNt-Zk

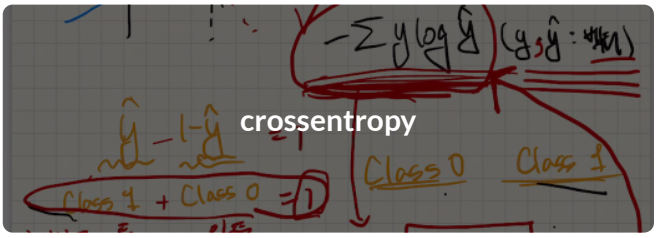
'practice_자료구조' 카테고리의 다른 글

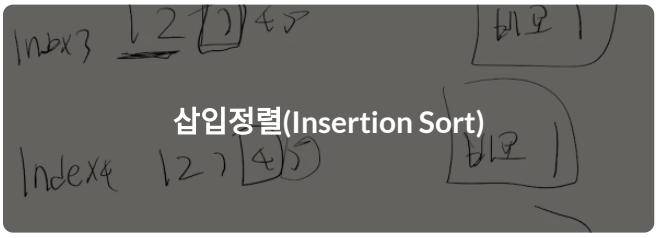
P / NP (0)	2024.09.30
해시 테이블 (Hash Table) (0)	2024.09.26
crossentropy (1)	2024.09.20
삽입정렬(Insertion Sort) (0)	2024.09.15

관련글 관련글 더보기









자연어(NLP)

네이쳐2024 님의 블로그입니다.

구독하기 +

댓글 0

이름

비밀번호

내용을 입력하세요.

등록