

삼성 청년 SW 아카데미

SW문제해결응용

<알림>

본 강의는 삼성 청년 SW아카데미의 콘텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

Day2. DFS 응용

DFS 응용

DFS는 완전 탐색 알고리즘으로 다양한 문제 유형이 존재한다.

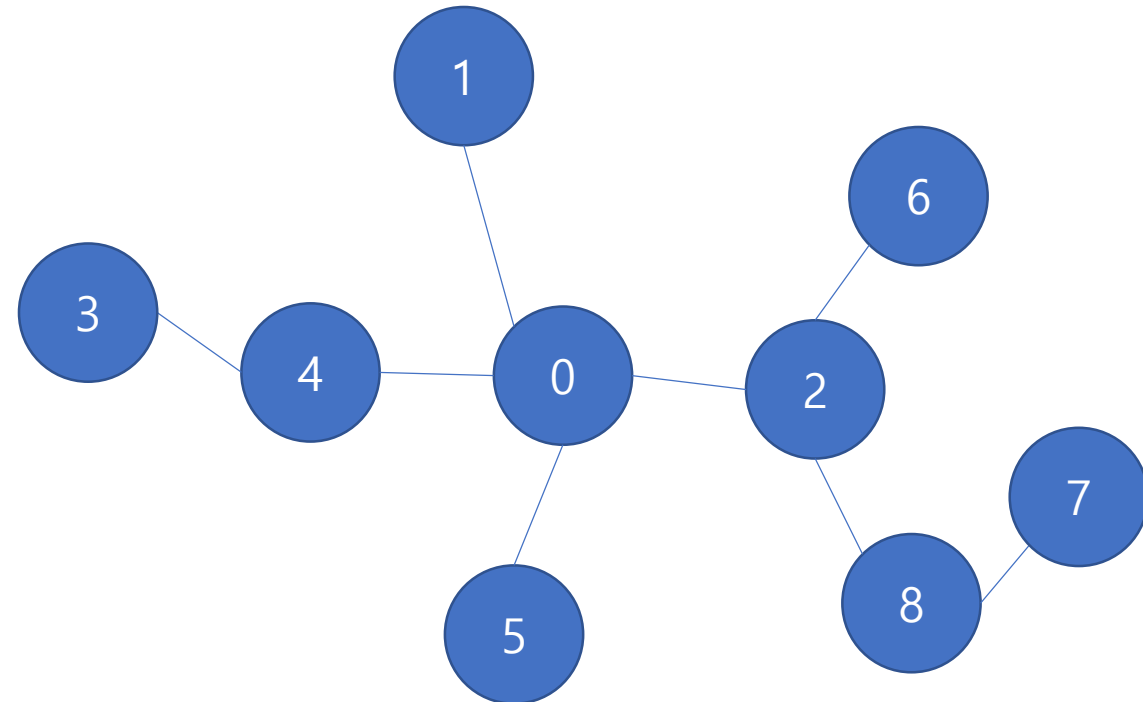
- 연결 요소 찾기
- 미로 탐색
- 사이클 탐지
- 위상 정렬
- 부분집합 생성
- 경로 찾기
- 섬의 개수
- 등등

그 중 BFS와 다르게, 특정 노드에서 다양한 경로 찾기 유형이 많이 출제된다!

DFS로 다음과 같은 작업을 순서대로 진행한다.

Confidential

1. 0번 노드에서 특정 노드까지 경로 뽑기
2. start 에서 goal까지 경로 뽑기
3. 그래프에서 가장 긴 경로 뽑기



1. 0번 노드에서 특정 노드까지 경로 뽑기

- 0번 노드부터 dfs를 시작한다.
- now==goal 이면 도착!
- 방문여부 체크
- 경로 기록 필요 (path 를 위한 level 사용)

<https://gist.github.com/hoconoco/15ad5a072bf07461307adb03abbd722e>

```
void dfs(int level, int now) {
    // 목표 노드(goal)에 도달하면 경로 출력
    if (now == goal) {
        for (int i = 0; i <= level; i++) {
            cout << path[i] << " ";
        }
        cout << "\n";
        return;
    }
    //갈 수 있는 노드 탐색
    for (int i = 0; i < arr[now].size(); i++) {
        int next = arr[now][i];
        //이미 방문했다면 넘어가
        if (visited[next] == 1) continue;
        //level+1 에 next를 기록한다! ( path[0]은 0번 노드의 자리 )
        path[level + 1] = next;
        visited[next] = 1;
        //다음 level 과 다음 노드로 탐색
        dfs(level + 1, next);
        visited[next] = 0;
        //경로 지우기
        path[level + 1] = 0;
        /*
        0으로 경로를 지우지만,
        0번 노드를 기록하기 때문에,
        -1과 같은 값으로 지워주는 것도 괜찮다.
        */
    }
}
```

2. start → goal 까지 경로 뽑기

- DFS는 특히 다양한 경로 문제에 적합하다.

```
int tc;
cin >> tc;

for (int i = 0; i < tc; i++) {
    //출발 노드와 도착 노드 입력
    int start;
    cin >> start >> goal;
    //path[0] 에 0번 노드를 현재 채웁하므로, 초기화를 -1로 하는 게 디버깅에 효과적이다.
    memset(path, -1, sizeof(path));
    memset(visited, 0, sizeof(visited));

    //0번 노드 방문 여부 체크
    visited[start] = 1;
    //path[0] 에 시작 노드 기록 -> 0번이 항상 맨 앞에 있다.
    path[0] = start;
    //dfs(level, now) : level : 탐색 깊이, now : 현재 방문 노드
    //start 부터 시작!
    dfs(0, start);
}
```

<https://gist.github.com/hoconoco/d44b206aa94d03092b59cf4c5a2775c7>

3. 그래프에서 가장 긴 경로 뽑기

- 모든 노드에서 dfs를 처리한다.

```
// 모든 노드에 대해 DFS 수행
for (int start = 0; start < N; start++) {
    visited[start] = 1; // 시작 노드 방문 처리
    path[0] = start; // 경로 시작
    dfs(0, start);
    visited[start] = 0; // 시작 노드 방문 처리 해제
}

// 가장 긴 경로 출력
cout << "가장 긴 경로: ";
for (int i = 0; i < longestPath.size(); i++) {
    cout << longestPath[i] << " ";
}
cout << "\n";
```

내일 방송에서 만나요!

삼성 청년 SW 아카데미