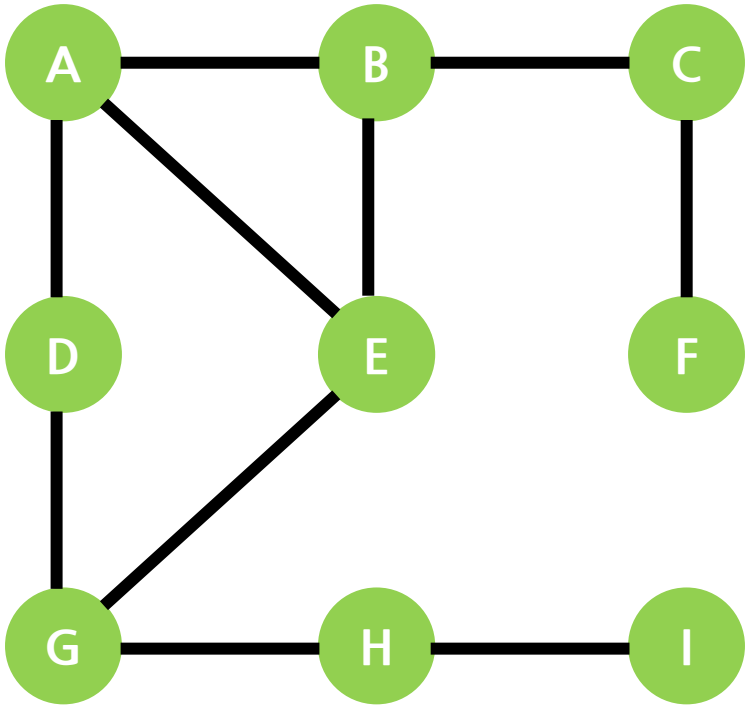
그래프 탐색 2가지 방법

- DFS (Depth-first search) 깊이 우선 탐색
- BFS (Breadth-first search) 넓이 우선 탐색

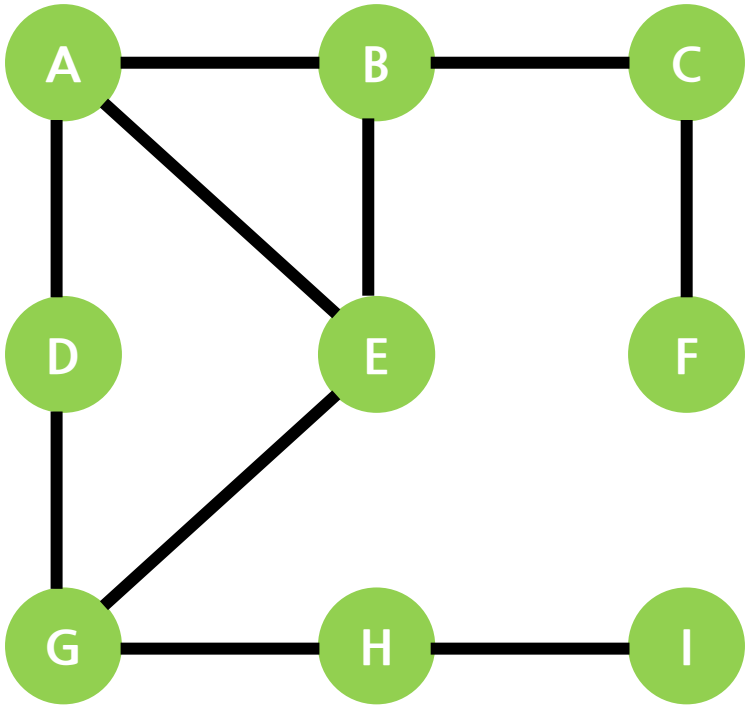
DFS (Depth-first search) 깊이 우선 탐색

스택 이용



BFS (Breadth-first search) 넓이 우선 탐색

큐 이용



DFS

모든 경우를 하나하나 다 탐색해야 할 경우에 사용

ex) 순열, 조합

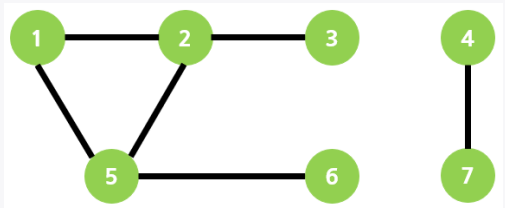
예제 입력 1 복사

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

노드 개수

엣지 개수

그래프 정보



인접리스트

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

스택 => 재귀함수로 구현

인접리스트 생성 후, 재귀적으로 순회

DFS

```cpp
#include <iostream>
#include <vector>

using namespace std;

vector<int> v[101]; //인접 리스트
bool check[101];
int ans = 0;

void dfs(int node) {
    check[node] = true;

    for (int i = 0; i < v[node].size(); i++) {
        int next = v[node][i];

        if (!check[next]) {
            dfs(next);
            ans++;
        }
    }
}
```

```cpp
int main() {
    int a, b; //a:노드 개수, b:엣지 개수

    cin >> a >> b;
    for (int i = 0; i < b; i++) {
        int tmp1, tmp2;

        cin >> tmp1 >> tmp2;

        /* 인접 리스트 생성 */
        v[tmp1].push_back(tmp2);
        v[tmp2].push_back(tmp1);
    }

    dfs(1);

    return 0;
}
```

BFS

depth 특징을 활용해야 하는 경우에 사용

ex) 최단경로

문제

N×M크기의 배열로 표현되는 미로가 있다.

1	0	1	1	1	1
1	0	1	0	1	0
1	0	1	0	1	1
1	1	1	0	1	1

cnt[x][y]

1		9	10	11	12
2		8		12	
3		7		13	14
4	5	6		14	15

BFS

```cpp
#include <iostream>
#include <queue>

using namespace std;

int N, M;
int map[100][100];
bool visited[100][100];
int cnt[100][100];

int dx[4] = { 0, 1, 0, -1 };
int dy[4] = { 1, 0, -1, 0 };

void bfs(int x, int y) {
    visited[x][y] = true;
    cnt[x][y]++; //시작 좌표 cnt 지정

    queue<pair<int, int>> q;
    q.push({ x,y });

    while (!q.empty()) {
        int xx = q.front().first;
        int yy = q.front().second;

        q.pop();
        for (int i = 0; i < 4; i++) {
            int nx = xx + dx[i];
            int ny = yy + dy[i];

            if (nx >= 0 && nx < N && ny >= 0 && ny < M && !visited[nx][ny] && map[nx][ny] == 1) {
                visited[nx][ny] = true;
                q.push({ nx,ny });
                cnt[nx][ny] = cnt[xx][yy] + 1;
            }
        }
    }
}
```

```cpp
int main() {

    bfs(0, 0);

    return 0;
}
```