

정보올림피아드 중급반

6주차

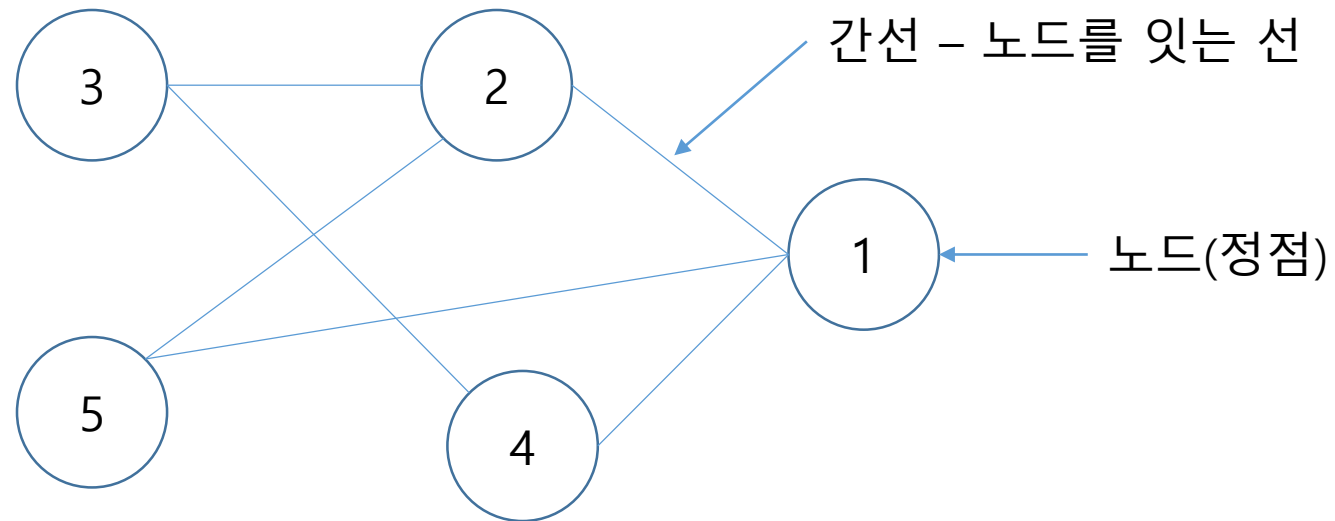
그래프

기본개념

그래프란

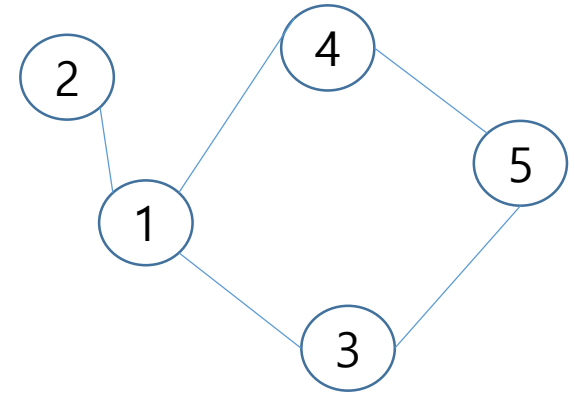
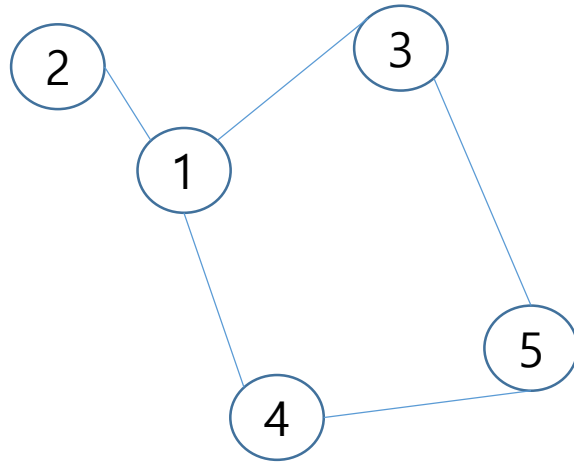
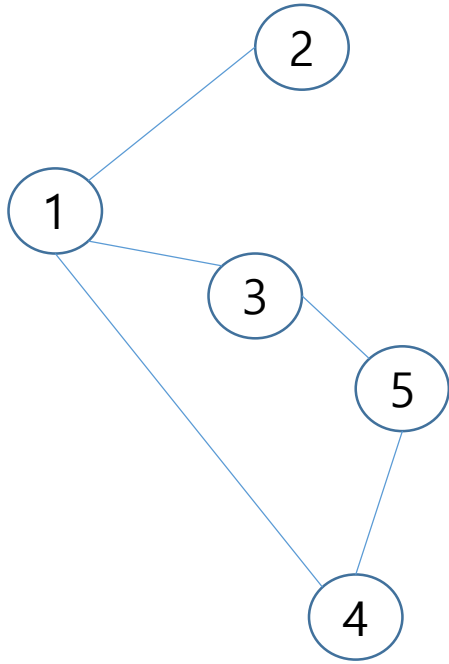
어떤 상호관계에 놓인 데이터를 표현하기 위해 고안된 자료구조
다양한 현실세계에 있는 문제들을 표현할 수 있다

Ex) 도시들을 연결하는 도로망, 사람들 간의 관계, 웹사이트의 링크 관계 등



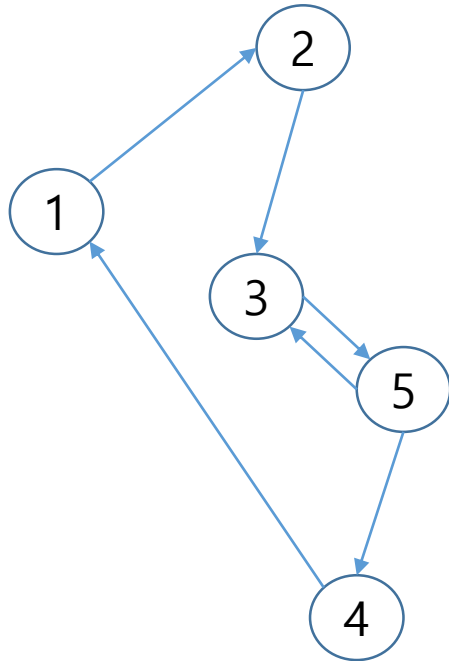
그래프란

그래프 $G(V,E)$ 는 정점들의 집합 (V)와 간선들의 집합(E)로 구성된다.
아래 3개의 그래프는 모두 같은 그래프이다.

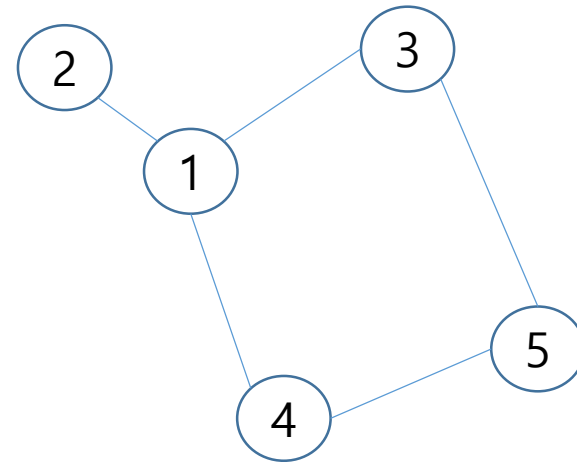


그래프의 종류

1. 방향 그래프 – 진행 방향 존재

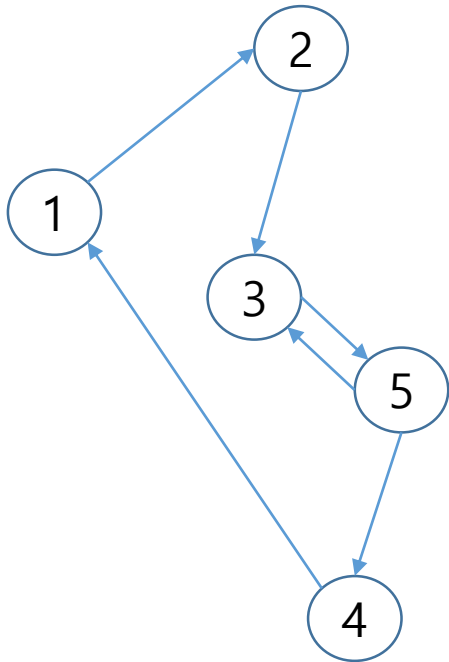


2. 무향 그래프



그래프의 경로

그래프에서 가장 중요한 개념 중 하나로, 끝과 끝이 연결된 간선들을 순서대로 나열한 것이다.



왼쪽 그래프에서
(1,2), (2,3), (3,5), (5,4)는 끝과 끝이 이어진 간선들이므로,
하나의 경로를 이룬다.

이걸 1-2-3-5-4 라고 표현합니다.
경로 중 한 정점을 최대 한번만 지나는 경로를 단순 경로(simple path)라고 한다.

왼쪽과 같이 1-2-3-5-4-1 로 시작점으로 끝나는 경로를 사이클이라고 부른다.

그래프의 모든 정점들을 특정한 순서에 따라 방문하는 알고리즘

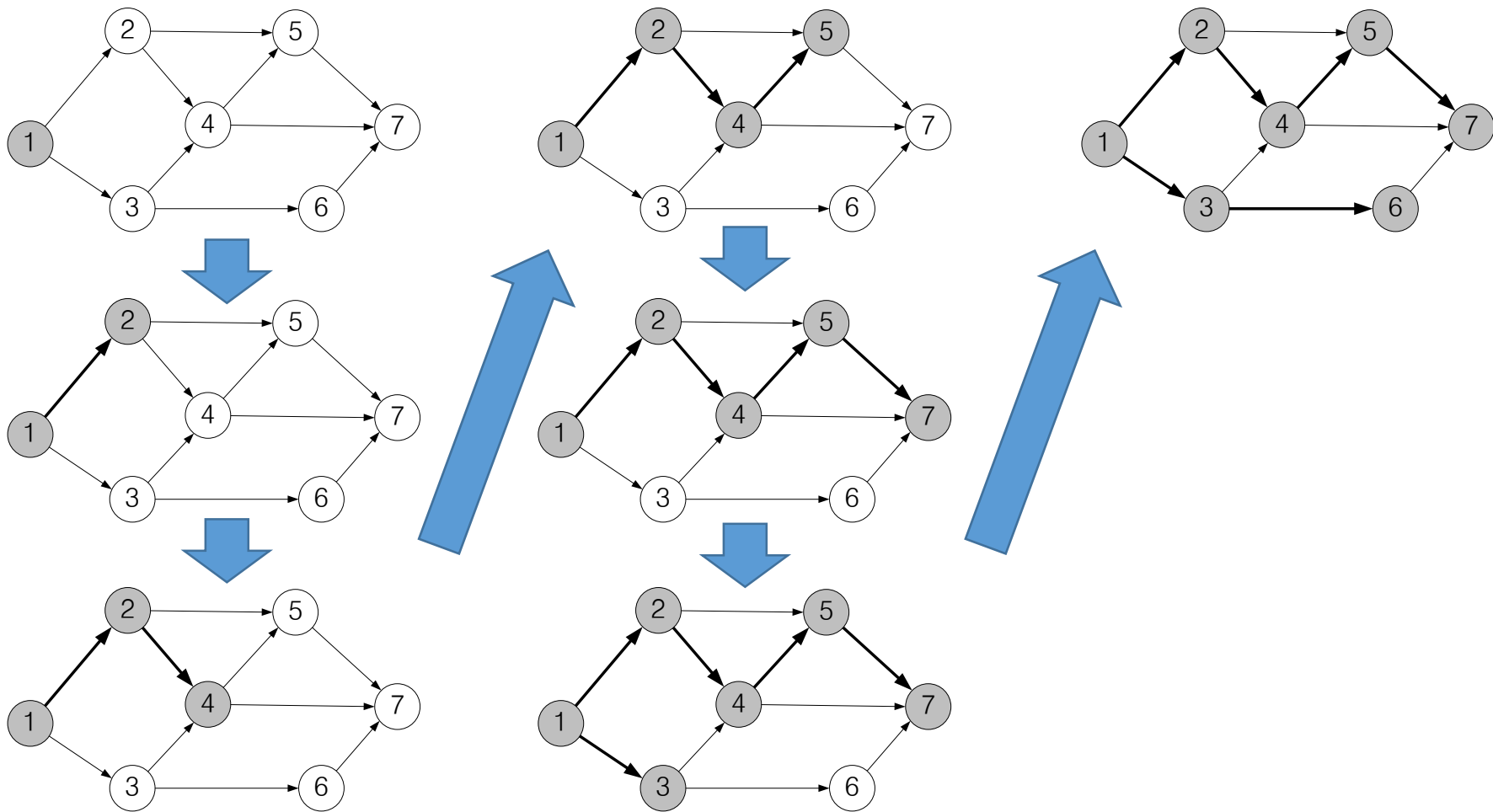
탐색을 하면서, 어떤 간선이 사용되었는지, 어떤 순서로 정점들이 방문 되었는 지를 통해 그래프의 구조를 알 수 있다.

대표적인 탐색 알고리즘 (단순 경로)

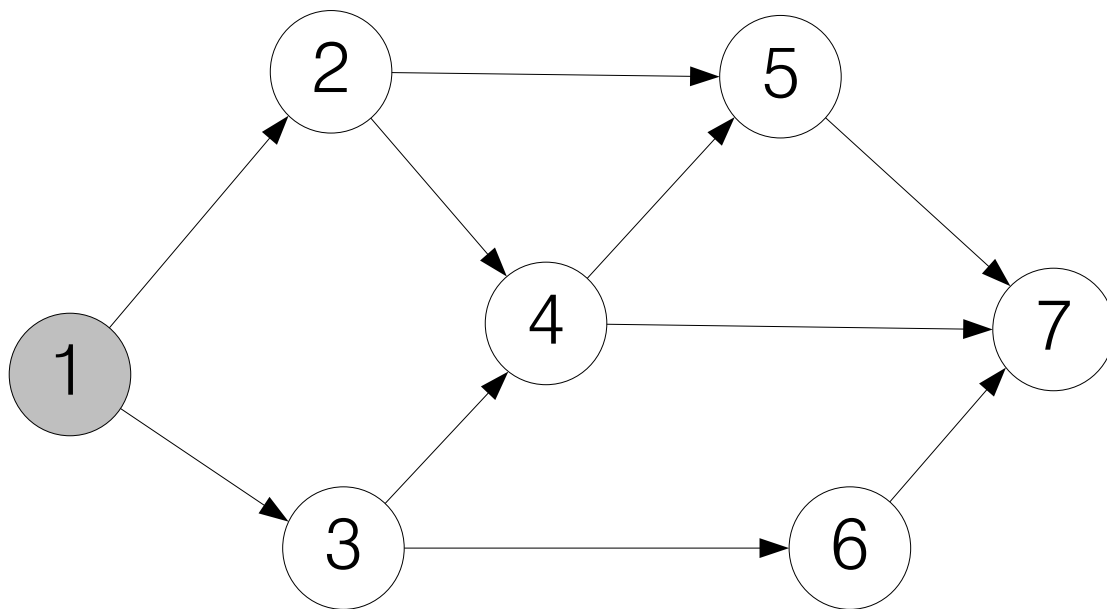
1. 깊이 우선 탐색(Depth-First Search, DFS)
2. 너비 우선 탐색(Breadth-First Search, BFS)

DFS(Depth First Search)

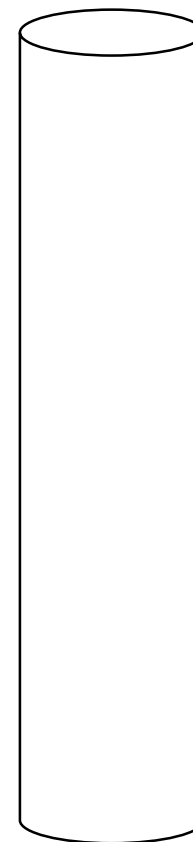
- 최대한 많이 깊숙히
- 스택을 이용해서 갈 수 있는 만큼 최대한 많이 가고 갈 수 없으면 이전 정점으로 돌아감
- 재귀 호출을 이용



DFS 탐색



1 -> 2
1 -> 3
2 -> 4
2 -> 5
3 -> 4
3 -> 6
4 -> 5
4 -> 7
5 -> 7

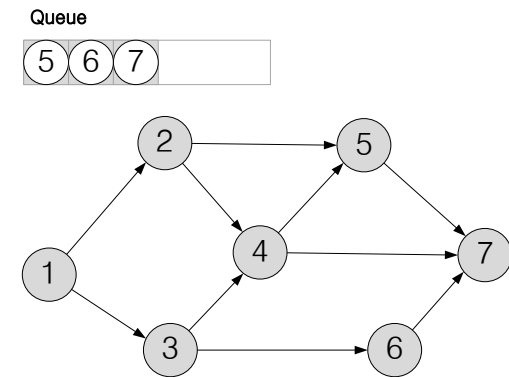
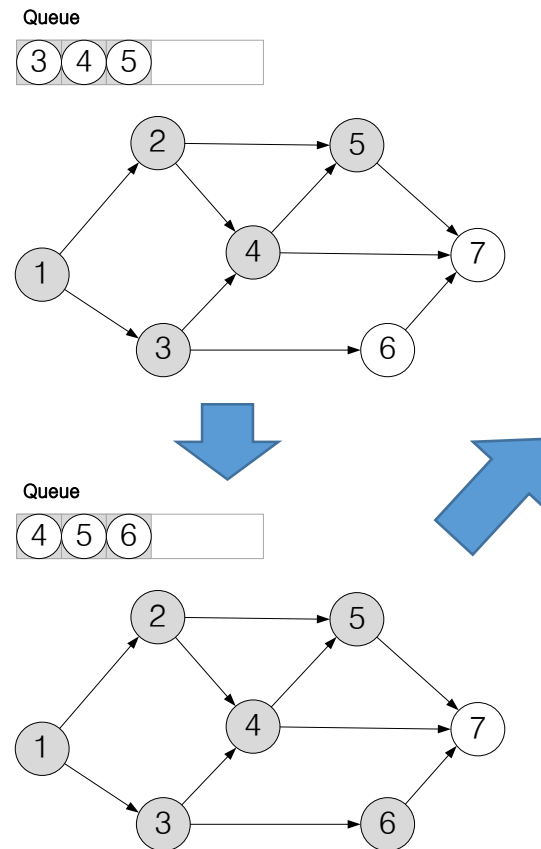
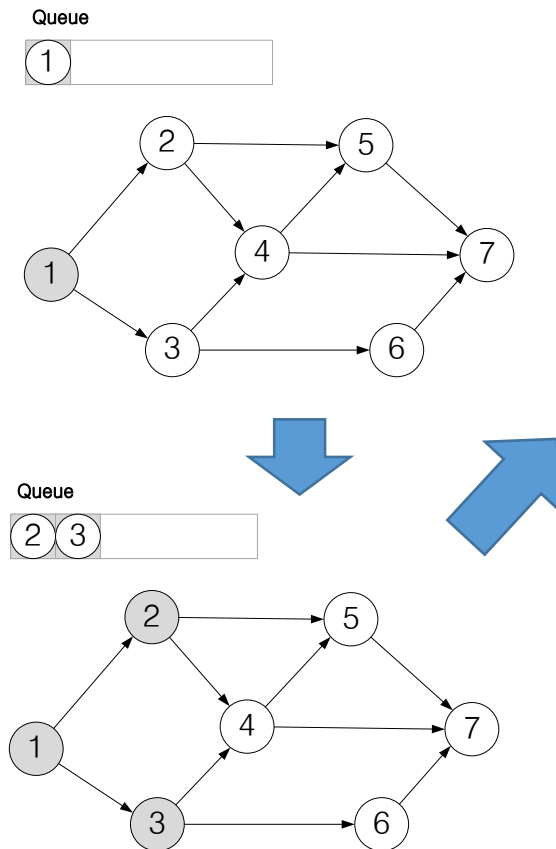


stack

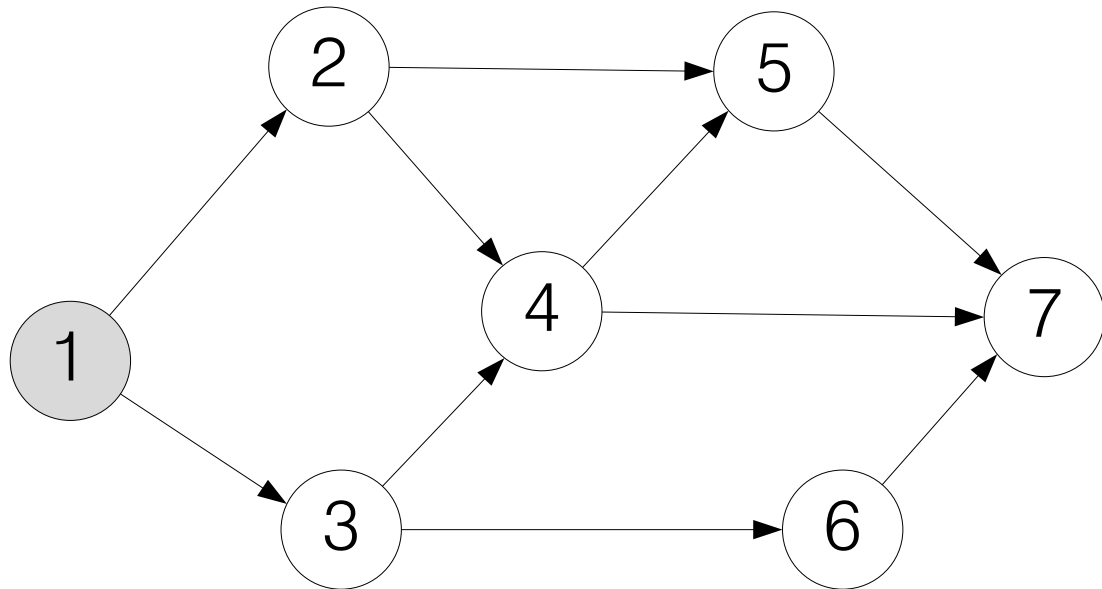
```
bool check[1001];  
void dfs(int node){  
    check[node] = true;  
    printf("%d",node);  
    for(int i=0; i<a[node].size(); i++){  
        int next = a[node][i];  
        if(check[next] == false)  
            dfs(next);  
    }  
}
```

BFS(Breadth First Search)

- 최대한 넓게 가는 것
- 지금 위치에서 갈 수 있는 모든 것을 큐에 넣음
- 모든 가중치 1 경우 최단 경로
- 반복문을 이용



BFS 탐색



queue

BFS 코드

```
void bfs(int start){
    queue <int> q;
    check[start] = true;
    q.push(start);
    while(!q.empty()){
        int node = q.front();
        q.pop();
        printf("%d ",node);
        for(int i=0; i<a[node].size(); i++){
            int next = a[node][i];
            if(check[next]==false){
                check[next] = true;
                q.push(next);
            }
        }
    }
}
```

[문제] DFS와 BFS

그래프를 DFS로 탐색한 결과와 BFS로 탐색한 결과를 출력하는 프로그램을 작성하시오. 단, 방문할 수 있는 정점이 여러 개인 경우에는 정점 번호가 작은 것을 먼저 방문하고, 더 이상 방문할 수 있는 점이 없는 경우 종료한다. 정점 번호는 1번부터 N번까지이다.

입력

첫째 줄에 정점의 개수 N ($1 \leq N \leq 1,000$), 간선의 개수 M ($1 \leq M \leq 10,000$), 탐색을 시작할 정점의 번호 V 가 주어진다. 다음 M 개의 줄에는 간선이 연결하는 두 정점의 번호가 주어진다. 어떤 두 정점 사이에 여러 개의 간선이 있을 수 있다. 입력으로 주어지는 간선은 양방향이다.

출력

첫째 줄에 DFS를 수행한 결과를, 그 다음 줄에는 BFS를 수행한 결과를 출력한다. V 부터 방문된 점을 순서대로 출력하면 된다.

예제 입력 1

4 5 1

1 2

1 3

1 4

2 4

3 4

예제 출력 1

1 2 4 3

1 2 3 4

소스 코드

```
#include <cstdio>
#include <algorithm>
#include <cstring>
#include <vector>
#include <queue>
using namespace std;
vector<int> a[1001];
bool check[1001];
```

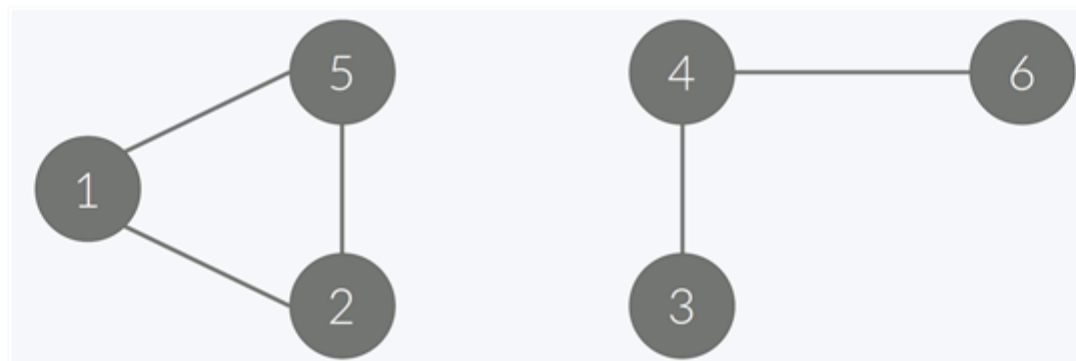
```
void dfs(int node) {
}

void bfs(int start) {
}
```

```
int main() {  
    int n, m, start;  
    scanf("%d %d %d",&n,&m,&start);  
    for (int i=0; i<m; i++) {  
        int u,v;  
        scanf("%d %d",&u,&v);  
        a[u].push_back(v);  
        a[v].push_back(u);  
    }  
    for (int i=1; i<=n; i++) {  
        sort(a[i].begin(), a[i].end());  
    }  
    dfs(start);  
    puts("");  
    bfs(start);  
    puts("");  
    return 0;  
}
```

[문제] 연결 요소의 개수

방향 없는 그래프가 주어졌을 때, 연결 요소 (Connected Component)의 개수를 구하는 프로그램을 작성하시오.



입력

첫째 줄에 정점의 개수 N 과 간선의 개수 M 이 주어진다. ($1 \leq N \leq 1,000$, $0 \leq M \leq N \times (N-1)/2$) 둘째 줄부터 M 개의 줄에 간선의 양 끝점 u 와 v 가 주어진다. ($1 \leq u, v \leq N$, $u \neq v$) 같은 간선은 한 번만 주어진다.

출력

첫째 줄에 연결 요소의 개수를 출력한다.

예제 입력 1

6 5

1 2

2 5

5 1

3 4

4 6

예제 출력 1

2

예제 입력 2

6 8

1 2

2 5

5 1

3 4

4 6

5 4

2 4

2 3

예제 출력 2

1

플러드 필 (Flood Fill)

[문제] 단지번호 붙이기

<그림 1>과 같이 정사각형 모양의 지도가 있다. 1은 집이 있는 곳을, 0은 집이 없는 곳을 나타낸다. 철수는 이 지도를 가지고 연결된 집들의 모임인 단지를 정의하고, 단지에 번호를 붙이려 한다. 여기서 연결되었다는 것은 어떤 집이 좌우, 혹은 아래위로 다른 집이 있는 경우를 말한다. 대각선상에 집이 있는 경우는 연결된 것이 아니다. <그림 2>는 <그림 1>을 단지별로 번호를 붙인 것이다. 지도를 입력하여 단지수를 출력하고, 각 단지에 속하는 집의 수를 오름차순으로 정렬하여 출력하는 프로그램을 작성하시오.

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

<그림 1>

0	1	1	0	2	0	0
0	1	1	0	2	0	2
1	1	1	0	2	0	2
0	0	0	0	2	2	2
0	3	0	0	0	0	0
0	3	3	3	3	3	0
0	3	3	3	0	0	0

<그림 2>

입력 형식

첫 번째 줄에는 지도의 크기 N (정사각형이므로 가로와 세로의 크기는 같으며 $5 \leq N \leq 25$)이 입력되고, 그 다음 N 줄에는 각각 N 개의 자료(0혹은 1)가 입력된다.

출력 형식

첫 번째 줄에는 총 단지수를 출력하시오. 그리고 각 단지내 집의 수를 오름차순으로 정렬하여 한 줄에 하나씩 출력하시오.

입력

7
0110100
0110101
1110101
0000111
0100000
0111110
0111000

출력

3
7
8
9