

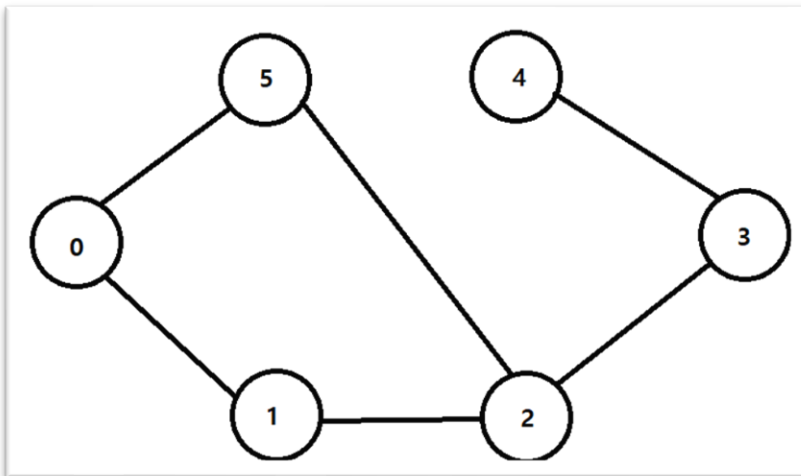
**CHAPTER 14**

**그래프 구현 및 응용**

# 실습 1

문제 1)

인접 리스트 방식으로 아래와 같은 그래프를 구현하고, 각 노드의 인접 리스트를 출력하시오. (무방향 그래프)



```
Microsoft Visual Studio 디버깅 콘솔
정점 0 의 인접 리스트 -> 5 -> 1
정점 1 의 인접 리스트 -> 2 -> 0
정점 2 의 인접 리스트 -> 5 -> 3 -> 1
정점 3 의 인접 리스트 -> 4 -> 2
정점 4 의 인접 리스트 -> 3
정점 5 의 인접 리스트 -> 2 -> 0
```

(뒷장에 구조체와 함수의 틀이 제시되어있음)

# 실습 1

문제 1-참고) 다음을 참고하여 코드를 작성하시오.

```
// 노드
typedef struct GraphNode {
    int vertex;                // 자신의 번호
    struct GraphNode* link;    // 연결된 다른 vertex의 주소
}GraphNode;

// 그래프
typedef struct GraphType {
    int n;                    // edge의 갯수
    GraphNode* adj_list[MAX_VERTICES]; // 인접 리스트
}GraphType;
```

```
// 그래프 초기화
void graph_init(GraphType *g) {
```

```
// vertex 추가
void insert_vertex(GraphType *g, int v) {
```

```
// edge 추가 (v를 u의 인접 리스트에 삽입)
void insert_edge(GraphType *g, int u, int v) {
```

```
// 인접 리스트 출력
void print_adj_list(GraphType* w, int i)
```

## 실습 2

문제 2)

문제 1에서 구현한 그래프를 너비 우선 탐색(BFS) 방식으로 순회하는 함수를 제작하시오. (큐 사용)

C# Microsoft Visual Studio 디버그 콘솔

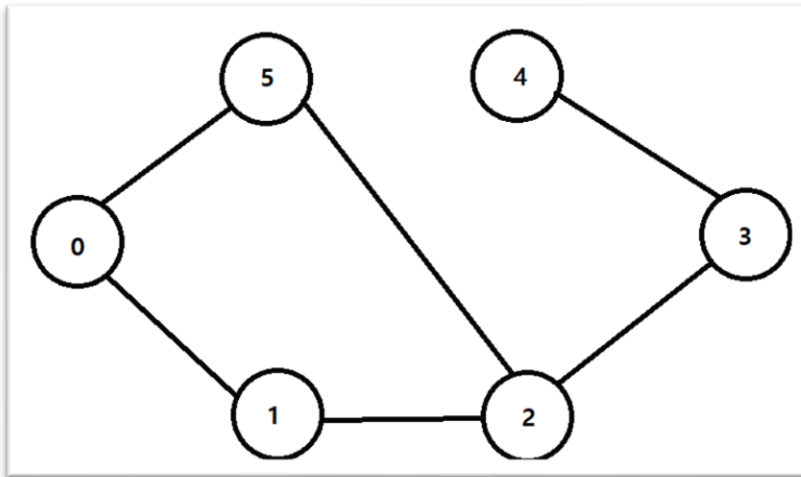
```
vertex 0 의 인접 리스트 -> 5 -> 1  
vertex 1 의 인접 리스트 -> 2 -> 0  
vertex 2 의 인접 리스트 -> 5 -> 3 -> 1  
vertex 3 의 인접 리스트 -> 4 -> 2  
vertex 4 의 인접 리스트 -> 3  
vertex 5 의 인접 리스트 -> 2 -> 0  
BFS 결과 : 0 -> 5 -> 1 -> 2 -> 3 -> 4
```

```
void bfs_list(GraphType* g, int v) {
```

## 실습 3

문제 3)

인접 행렬 방식으로 아래와 같은 그래프를 구현하고,  
인접 행렬을 출력하시오. (무방향 그래프)



선택 Microsoft Visual Studio 디버그 콘솔

인접 행렬 출력 결과

	0	1	2	3	4	5
0	0	1	0	0	0	1
1	1	0	1	0	0	0
2	0	1	0	1	0	1
3	0	0	1	0	1	0
4	0	0	0	1	0	0
5	1	0	1	0	0	0

(뒷장에 구조체와 함수의 틀이 제시되어있음)

## 실습 4

문제 4)

문제 3에서 구현한 그래프를 깊이 우선 탐색(DFS) 방식으로 순회하는 함수를 제작하시오. (스택 사용)

```
Microsoft Visual Studio 디버그 콘솔
인접 행렬 출력 결과
  | 0 1 2 3 4 5
--+-----+
0 | 0 1 0 0 0 1
1 | 1 0 1 0 0 0
2 | 0 1 0 1 0 1
3 | 0 0 1 0 1 0
4 | 0 0 0 1 0 0
5 | 1 0 1 0 0 0

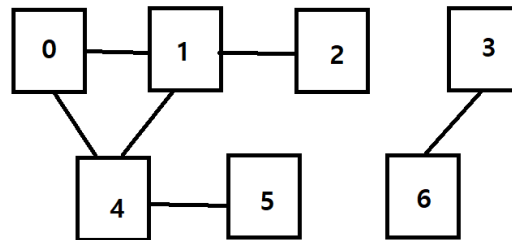
DFS 결과 (시작 정점 : 3)
-> 3 -> 2 -> 1 -> 0 -> 5 -> 4
```

```
void dfs_mat(GraphType* g, int v) {
```

## 실습 5

문제 5)

7대의 컴퓨터가 아래 그림과 같이 네트워크 상에서 연결되어 있다고 하자. 0번 컴퓨터가 웜 바이러스에 걸리면 웜 바이러스는 1번과 4번 컴퓨터를 거쳐 2번과 5번 컴퓨터까지 전파되어 1, 2, 4, 5 네 대의 컴퓨터는 웜 바이러스에 걸리게 된다. 하지만 3번과 6번 컴퓨터는 1번 컴퓨터와 네트워크상에서 연결되어 있지 않기 때문에 영향을 받지 않는다.



어느 날, 0번 컴퓨터가 웜 바이러스에 걸렸다면, 0번 컴퓨터를 통해 웜 바이러스에 걸리게 되는 컴퓨터의 수를 출력하는 프로그램을 작성하시오. (0번 컴퓨터는 제외)

# 실습 5

문제 5-예시)

(입력, 프로그램 내부)

```
// edge 추가
insert_edge(g, 0, 1);
insert_edge(g, 0, 4);
insert_edge(g, 1, 2);
insert_edge(g, 4, 5);
insert_edge(g, 3, 6);

insert_edge(g, 1, 0);
insert_edge(g, 4, 0);
insert_edge(g, 2, 1);
insert_edge(g, 5, 4);
insert_edge(g, 6, 3);
```

출력(콘솔)

감염된 컴퓨터의 갯수 : 4



## 실습 6

문제 6)

$N * M$  크기의 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

1: 이동할 수 있는 칸

0: 이동할 수 없는 칸

이러한 미로가 주어졌을 때, (1, 1)에서 출발하여 (N, M)의 위치로 이동할 때 지나야 하는 최소의 칸 수를 구하는 프로그램을 작성하시오. (대각선 이동 X)

힌트 : BFS를 이용한다.

# 실습 6

## 문제 6) 입출력 예시

```
Microsoft Visual Studio
4 6
101111
101010
101011
111011
15칸을 지나야 함.
```

```
Microsoft Visual Studio
4 6
110110
110110
111111
111101
9칸을 지나야 함.
```

```
Microsoft Visual Studio 디버그 콘솔
2 25
1011101110111011101110111
1110111011101110111011101
38칸을 지나야 함.
```

```
Microsoft Visual Studio
7 7
1011111
1110001
1000001
1000001
1000001
1000001
1111111
13칸을 지나야 함.
```

(입력)  
첫째 줄 : N M  
둘째 줄 ~ : 미로

(출력)  
(숫자)칸을 지나야 함.

# 질문 및 정리

