

10장 그래프 #2(인접리스트를 이용해 구현한)

```
//인접리스트그래프.c 소스파일
#include <stdio.h>
#include <stdlib.h>2
#include "queue.h" // <-----

#define TRUE 1
#define FALSE 0
#define MAX_VERTICES 50 // 디버깅할 때는 작은 수로 설정해서 하는 것이 좋습니다!!

int visited[MAX_VERTICES];

typedef struct GraphNode
{
    int vertex;
    struct GraphNode *link;
} GraphNode;

typedef struct GraphType {
    int n; // 정점의 개수
    GraphNode *adj_list[MAX_VERTICES];
} GraphType;

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

}

// 정점 삽입 연산
void insert_vertex(GraphType *g, int v)
{
    if( ((g->n)+1) > MAX_VERTICES ){
        fprintf(stderr, "그래프: 정점의 개수 초과");
        return;
    }
    g->n++;
}

// 간선 삽입 연산, v를 u의 인접 리스트에 삽입한다.
void insert_edge(GraphType *g, int u, int v)
{
    GraphNode *node;
    if( u >= g->n || v >= g->n ){
        fprintf(stderr, "그래프: 정점 번호 오류");
        return;
    }

    // u 에 v를 매단다

    // v 에 u를 매단다

}
```

```
/* 깊이 우선 탐색 depth_first_search(v)
v를 방문되었다고 표시;
for all u ∈ (v에 인접한 정점) do
if (u 가 아직 방문되지 않았으면)
then depth_first_search(u) */
void dfs_list(GraphType *g, int v) // 깊이 우선 탐색(인접 리스트)
{
    GraphNode *w;
    visited[v] = TRUE; // 정점 v의 방문 표시
    printf("%d ", v); // 방문한 정점 출력

}

/* 넓이 우선 탐색 breadth_first_search(v)
v를 방문되었다고 표시;
큐 Q에 정점 v를 삽입;
while (not is_empty(Q)) do
    큐 Q에서 정점 w를 삭제;
    for all u ∈ (w에 인접한 정점) do
        if (u 가 아직 방문되지 않았으면) then u를 방문되었다고 표시;
        u를 큐 Q에 삽입; */
void bfs_list(GraphType *g, int v) // 너비 우선 탐색(인접 리스트)
{
    GraphNode *w;
    QueueType q;
    init(&q); // 큐 초기 화
    visited[v] = TRUE; // 정점 v 방문 표시
    printf("%d ", v);
    enqueue(&q, v); // 시작정점을 큐에 저장
    while(!is_empty(&q)){
        v = dequeue(&q); // 큐에 저장된 정점 선택

    }
}

void visited_init()
{
    int i;
    for (i = 0; i < MAX_VERTICES; i++)
        visited[i] = 0;
}
```

<pre>int main(void) { GraphType g; graph_init(&g); insert_vertex(&g, 0); insert_vertex(&g, 1); insert_vertex(&g, 2); insert_vertex(&g, 3); insert_edge(&g, 0, 1); insert_edge(&g, 1, 2); insert_edge(&g, 2, 3); insert_edge(&g, 3, 0); insert_edge(&g, 0, 2);</pre>	<pre>printf("₩n 깊이 우선 탐색₩n"); dfs_list(&g, 0); printf("₩n"); visited_init(); dfs_list(&g, 1); printf("₩n"); visited_init(); dfs_list(&g, 2); printf("₩n"); visited_init(); dfs_list(&g, 3); printf("₩n");</pre>	<pre>printf("₩n 넓이 우선 탐색₩n"); visited_init(); bfs_list(&g, 0); printf("₩n"); visited_init(); bfs_list(&g, 1); printf("₩n"); visited_init(); bfs_list(&g, 2); printf("₩n"); visited_init(); bfs_list(&g, 3); printf("₩n");</pre>
---	--	--