

자료구조 실습과제 14

- 솔루션 및 프로젝트 명칭 : Proj_14_이름이니셜
- 제출방법 : 아래 문제를 해결하기 위한 프로그램을 구현한 후 컴파일 및 실행한 후, 오류가 없으면 메뉴에서 솔루션 정리를 수행한 후 윈도우 탐색기에서 솔루션 폴더를 찾아 압축하여 E-class에 올림

각 문제를 테스트하기 위하여 아래와 같이 구현하시오.

```
#define _CRT_SECURE_NO_WARNINGS

#define PROB 1 // 각각의 문제를 구현하고 해당 문제 번호를 변경하여 테스트

#if PROB == 1
// 1번 문제를 해결하기 위한 프로그램 작성

#elif PROB == 2
// 2번 문제를 해결하기 위한 프로그램 작성

#endif
```

문제 1) 인접행렬을 이용한 넓이 우선 탐색, 깊이 우선 탐색 구현

그래프는 정점과 정점 간의 연결선인 간선의 집합으로 구성된다. 방향성이 없는 연결된 그래프를 대상으로 깊이 우선 탐색과 너비 우선 탐색 알고리즘을 구현하시오. 단, 그래프는 인접행렬로 구현된다. 아래 그래프 정보를 저장하기 위한 구조체를 참고하여 인접행렬에 간선 정보를 저장한다. 아래 메인 함수에서 그래프 정보를 저장하는 내용과 탐색을 위한 함수 호출 방법을 참고하여 구현하시오.

```
#define MAX_VERTICES 50
typedef struct GraphType {
    int n; // 정점의 개수
    int adj_mat[MAX_VERTICES][MAX_VERTICES];
} GraphType;

int main(void)
{
    GraphType* g;
    int i;
    int visited[MAX_VERTICES];
```

```

g = (GraphType*)malloc(sizeof(GraphType));
init(g);

//그래프 초기화 및 간선 추가
set_vertex(g, 11);
insert_edge(g, 0, 1);
insert_edge(g, 0, 2);
insert_edge(g, 0, 3);
insert_edge(g, 1, 2);
insert_edge(g, 1, 3);
insert_edge(g, 2, 3);
insert_edge(g, 2, 4);
insert_edge(g, 4, 5);
insert_edge(g, 4, 7);
insert_edge(g, 4, 8);
insert_edge(g, 5, 6);
insert_edge(g, 6, 7);
insert_edge(g, 6, 10);
insert_edge(g, 7, 9);
insert_edge(g, 7, 10);
insert_edge(g, 8, 9);

//print matrix of graph
printf_graph_matrix(g);

//깊이 우선 탐색
for (i = 0; i < MAX_VERTICES; i++) visited[i] = 0;
printf("깊이 우선 탐색\n");
dfs_mat(g, 0, visited);
printf("\n\n");

//너비 우선 탐색
for (i = 0; i < MAX_VERTICES; i++) visited[i] = 0;
printf("너비 우선 탐색\n");
bfs_mat(g, 0, visited);
printf("\n");

free(g);
return 0;
}

```

실행 예 : 실행 예는 프로그램에서 다음 방문 노드를 선택하는 방식에 따라 결과가 다를 수 있음

```
Microsoft Visual Studio 디버그 콘솔

그래프 행렬 출력
  0 1 2 3 4 5 6 7 8 9 10
0 0 1 1 1 0 0 0 0 0 0
1 1 0 1 1 0 0 0 0 0 0
2 1 1 0 1 1 0 0 0 0 0
3 1 1 1 0 0 0 0 0 0 0
4 0 0 1 0 0 1 0 1 1 0 0
5 0 0 0 0 1 0 1 0 0 0 0
6 0 0 0 0 0 1 0 1 0 0 1
7 0 0 0 0 1 0 1 0 0 1 1
8 0 0 0 0 1 0 0 0 0 1 0
9 0 0 0 0 0 0 0 1 1 0 0
10 0 0 0 0 0 0 1 1 0 0 0

깊이 우선 탐색
[0] -> [1] -> [2] -> [3] -> [4] -> [5] -> [6] -> [7] -> [9] -> [8] -> [10] ->

너비 우선 탐색
[0] -> [1] -> [2] -> [3] -> [4] -> [5] -> [7] -> [8] -> [6] -> [9] -> [10] ->

D:\황형석\2023-01 강의\2023-자료구조\실습과제\So1_14\64\Debug\Proj.exe (프로세스 11636개)이<가> 종료되었습니다<코드: 0>
이 창을 닫으려면 아무 키나 누르세요...
```

문제 2) 인접리스트를 이용한 넓이 우선 탐색, 깊이 우선 탐색 구현

그래프는 정점과 정점 간의 연결선인 간선의 집합으로 구성된다. 방향성이 없는 연결된 그래프를 대상으로 깊이 우선 탐색과 너비 우선 탐색 알고리즘을 구현하시오. 단, 그래프는 인접리스트로 구현된다. 아래 그래프 정보를 저장하기 위한 구조체를 참고하여 인접리스트에 간선 정보를 저장한다. 아래 메인 함수에서 그래프 정보를 저장하는 내용과 탐색을 위한 함수 호출 방법을 참고하여 구현하시오.

```
#define MAX_VERTICES 50
typedef struct GraphNode
{
    int vertex;
    struct GraphNode* link;
} GraphNode;

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

int main(void)
{
    GraphType* g;
    int i;
    int visited[MAX_VERTICES];

    g = (GraphType*)malloc(sizeof(GraphType));
    init(g);

    //그래프 초기화 및 간선 추가
    set_vertex(g, 11);
    insert_edge(g, 0, 1);
    insert_edge(g, 0, 2);
    insert_edge(g, 0, 3);
    insert_edge(g, 1, 2);
    insert_edge(g, 1, 3);
    insert_edge(g, 2, 3);
    insert_edge(g, 2, 4);
    insert_edge(g, 4, 5);
    insert_edge(g, 4, 7);
    insert_edge(g, 4, 8);
    insert_edge(g, 5, 6);
    insert_edge(g, 6, 7);
    insert_edge(g, 6, 10);
    insert_edge(g, 7, 9);
    insert_edge(g, 7, 10);
    insert_edge(g, 8, 9);

    // 그래프 출력
    print_adj_list(g);

    // 깊이 우선 탐색
```

```

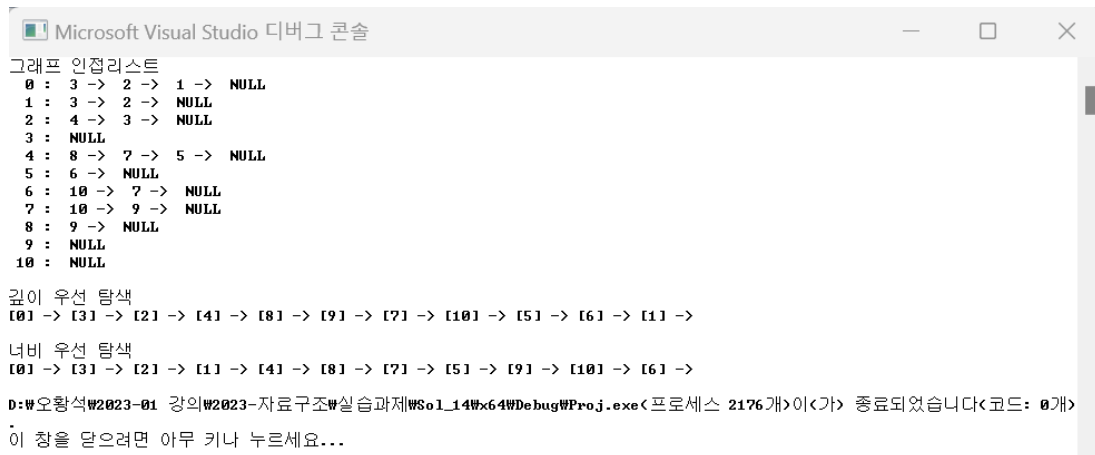
for (i = 0; i < MAX_VERTICES; i++) visited[i] = 0;
printf("깊이 우선 탐색\n");
dfs_list(g, 0, visited);
printf("\n\n");

// 너비 우선 탐색
for (i = 0; i < MAX_VERTICES; i++) visited[i] = 0;
printf("너비 우선 탐색\n");
bfs_list(g, 0, visited);
printf("\n");

free(g);
return 0;
}

```

실행 예 : 실행 예는 프로그램에서 다음 방문 노드를 선택하는 방식에 따라 결과가 다를 수 있음



The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```

그래프 인접리스트
0 : 3 -> 2 -> 1 -> NULL
1 : 3 -> 2 -> NULL
2 : 4 -> 3 -> NULL
3 : NULL
4 : 8 -> 7 -> 5 -> NULL
5 : 6 -> NULL
6 : 10 -> 7 -> NULL
7 : 10 -> 9 -> NULL
8 : 9 -> NULL
9 : NULL
10 : NULL

깊이 우선 탐색
[0] -> [3] -> [2] -> [4] -> [8] -> [9] -> [7] -> [10] -> [5] -> [6] -> [1] ->

너비 우선 탐색
[0] -> [3] -> [2] -> [1] -> [4] -> [8] -> [7] -> [5] -> [9] -> [10] -> [6] ->

D:\황석\2023-01 강의\2023-자료구조\실습과제\So1_14\64\Debug\Proj.exe <프로세스 2176개>이 <가> 종료되었습니다 <코드: 0개>
이 창을 닫으려면 아무 키나 누르세요...

```