

자료구조 실습과제 16

- 솔루션 및 프로젝트 명칭 : Proj_16_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

#elif PROB == 3
// 3번 문제를 해결하기 위한 프로그램 작성

#endif
```

문제 1) 두 정점간 최단경로를 찾는 Dijkstra 알고리즘

Dijkstra 알고리즘을 활용하여 그래프에서 시작 정점에서 특정 정점까지의 최단거리를 구하는 프로그램을 구현하시오. 그래프는 인접행렬로 표시되며, 각 에지의 가중치가 행렬 값으로 주어진다. INF로 설정된 경우는 두 정점간 직접적인 연결이 없는 경우이며, 그래프 에지의 가중치보다 매우 큰 값으로 할당하여 사용한다. 시작 정점에서 모든 정점에 이르는 최단거리를 구하고, 특정 정점을 입력하였을 경우 그 정점에 이르는 경로를 출력하시오. 실행 예는 아래와 같다. 만약 특정 정점에 이르는 경로가 존재하지 않을 경우 경로 없음이라고 출력한다.

```
#define TRUE 1
#define FALSE 0
#define MAX_VERTICES 100
#define INF 10000 /* 무한대 (연결이 없는 경우) */

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

int distance[MAX_VERTICES]; /* 시작정점으로부터의 최단경로 거리 */
int found[MAX_VERTICES]; /* 방문한 정점 표시 */
int path[MAX_VERTICES]; /* 경유 노드 */

int main(void)
{
    int end;

    // test data 1
    /*
    GraphType g = { 7,
    {{ 0, 7, INF, INF, 3, 10, INF },
    { 7, 0, 4, 10, 2, 6, INF },
    { INF, 4, 0, 2, INF, INF, INF },
    { INF, 10, 2, 0, 11, 9, 4 },
    { 3, 2, INF, 11, 0, INF, 5 },
    { 10, 6, INF, 9, INF, 0, INF },
    { INF, INF, INF, 4, 5, INF, 0 } }
    };
    */

    // 방향성 그래프에도 적용 가능
    // test data 2
    /*
    GraphType g = {6,
        {{ 0, 50, 45, 10, INF, INF},
        { INF, 0, 10, 15, INF, INF},
        { INF, INF, 0, INF, 30, INF},
        { 20, INF, INF, 0, 15, INF},
        { INF, 20, 35, INF, 0, INF},
        { INF, INF, INF, INF, 3, 0}}
    };
    */
}
```

```

// test data 3
/**
GraphType g = { 7,
    {{ 0, 7, INF, 5, INF, INF, INF},
      { 7, 0, 8, 9, 7, INF, INF},
      { INF, 8, 0, INF, 5, INF, INF},
      { 5, 9, INF, 0, 15, 6, INF },
      { INF, 7, 5, 15, 0, 8, 9},
      { INF, INF, INF, 6, 8, 0, 11 },
      { INF, INF, INF, INF, 9, 11, 0 } }
};
/**/

shortest_path(&g, 0);
end = 5;
printf("\n정점 %d에서 정점 %d까지의 경로 : ", 0, end);
print_path(&g, 0, end); // 정점 0에서 정점 end으로 가는 최단경로 출력
printf("\n");

return 0;
}

```

실행 예 : (test data1)

```

Microsoft Visual Studio 디버그 콘솔

STEP 1: distance: 0 7 * * 3 10 *
found: 1 0 0 0 0 0 0

STEP 2: distance: 0 5 * 14 3 10 8
found: 1 0 0 0 1 0 0

STEP 3: distance: 0 5 9 14 3 10 8
found: 1 1 0 0 1 0 0

STEP 4: distance: 0 5 9 12 3 10 8
found: 1 1 0 0 1 0 1

STEP 5: distance: 0 5 9 11 3 10 8
found: 1 1 1 0 1 0 1

STEP 6: distance: 0 5 9 11 3 10 8
found: 1 1 1 0 1 1 1

STEP 7: distance: 0 5 9 11 3 10 8
found: 1 1 1 1 1 1 1

Print Path Array
Index : 0 1 2 3 4 5 6
Path[]: 0 4 1 2 0 0 4

정점 0에서 정점 6까지의 경로 : <0->4> <4->6>

D:\오환석\2023-01 강의\자료구조\실습과제\Sol_19\Debug\Proj.exe<프로세스 29112개>이<가> 종료되었습니다.<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...

```

실행 예 : (test data2)

```

Microsoft Visual Studio 디버그 콘솔

STEP 1: distance: 0 50 45 10 * *
        found:    1 0 0 0 0 0

STEP 2: distance: 0 50 45 10 25 *
        found:    1 0 0 1 0 0

STEP 3: distance: 0 45 45 10 25 *
        found:    1 0 0 1 1 0

STEP 4: distance: 0 45 45 10 25 *
        found:    1 1 0 1 1 0

STEP 5: distance: 0 45 45 10 25 *
        found:    1 1 1 1 1 0

STEP 6: distance: 0 45 45 10 25 *
        found:    1 1 1 1 1 1

Print Path Array
Index : 0 1 2 3 4 5
Pathfil: 0 4 0 0 3 0

정점 0에서 정점 1까지의 경로 : <0->3> <3->4> <4->1>

D:\#오황석\W2023-01 강의\W2023-자료구조\실습과제\Sol_19\WDebug\Proj.exe<프로세스 18784개>이<가> 종료되었습니다<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...

```

```

Microsoft Visual Studio 디버그 콘솔

STEP 1: distance: 0 50 45 10 * *
        found:    1 0 0 0 0 0

STEP 2: distance: 0 50 45 10 25 *
        found:    1 0 0 1 0 0

STEP 3: distance: 0 45 45 10 25 *
        found:    1 0 0 1 1 0

STEP 4: distance: 0 45 45 10 25 *
        found:    1 1 0 1 1 0

STEP 5: distance: 0 45 45 10 25 *
        found:    1 1 1 1 1 0

STEP 6: distance: 0 45 45 10 25 *
        found:    1 1 1 1 1 1

Print Path Array
Index : 0 1 2 3 4 5
Pathfil: 0 4 0 0 3 0

정점 0에서 정점 5까지의 경로 : 경로 없음

D:\#오황석\W2023-01 강의\W2023-자료구조\실습과제\Sol_19\WDebug\Proj.exe<프로세스 2620개>이<가> 종료되었습니다<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...

```

실행 예 : (test data3)

```
Microsoft Visual Studio 디버그 콘솔

STEP 1: distance: 0 7 * 5 * * *
          found:  1 0 0 0 0 0 0

STEP 2: distance: 0 7 * 5 20 11 *
          found:  1 0 0 1 0 0 0

STEP 3: distance: 0 7 15 5 14 11 *
          found:  1 1 0 1 0 0 0

STEP 4: distance: 0 7 15 5 14 11 22
          found:  1 1 0 1 0 1 0

STEP 5: distance: 0 7 15 5 14 11 22
          found:  1 1 0 1 1 1 0

STEP 6: distance: 0 7 15 5 14 11 22
          found:  1 1 1 1 1 1 0

STEP 7: distance: 0 7 15 5 14 11 22
          found:  1 1 1 1 1 1 1

Print Path Array
Index : 0 1 2 3 4 5 6
Pathfil: 0 0 1 0 1 3 5

정점 0에서 정점 5까지의 경로 : <0->3> <3->5>

D:\오형석\2023-01 강의\2023-자료구조\실습과제\Sol_19\Debug\Proj.exe<프로세스 29728개>이<가> 종료되었습니다<코드: 0개>.
이 창을 닫으려면 아무 키나 누르세요...
```

문제 2) 모든 정점간 최단경로를 찾는 Floyd 알고리즘

그래프의 모든 정점간 최단거리를 구하는 Floyd 알고리즘을 구현하시오. 각 단계별 가중치 행렬의 값을 출력하시오. 모든 정점간의 최단 거리를 구한 경우 특정 두 정점 간의 경로를 출력하시오.

```
#define TRUE 1
#define FALSE 0
#define MAX_VERTICES 100
#define INF 1000000 /* 무한대 (연결이 없는 경우) */

typedef struct GraphType {
    int n; // 정점의 개수
    int weight[MAX_VERTICES][MAX_VERTICES];
} GraphType;

int A[MAX_VERTICES][MAX_VERTICES];
int path[MAX_VERTICES][MAX_VERTICES];

int main(void)
{
    int start, end;
    /*
    // test data 1
    GraphType g = { 7,
    {{ 0, 7, INF, INF, 3, 10, INF },
    { 7, 0, 4, 10, 2, 6, INF },
    { INF, 4, 0, 2, INF, INF, INF },
    { INF, 10, 2, 0, 11, 9, 4 },
    { 3, 2, INF, 11, 0, INF, 5 },
    { 10, 6, INF, 9, INF, 0, INF },
    { INF, INF, INF, 4, 5, INF, 0 } }
    };

    // test data 2
    GraphType g = { 7,
    {{ 0, 7, INF, 5, INF, INF, INF },
    { 7, 0, 8, 9, 7, INF, INF },
    { INF, 8, 0, INF, 5, INF, INF },
    { 5, 9, INF, 0, 15, 6, INF },
    { INF, 7, 5, 15, 0, 8, 9 },
    { INF, INF, INF, 6, 8, 0, 11 },
    { INF, INF, INF, INF, 9, 11, 0 } }
    };

    floyd(&g);
    start = 0; end = 5;
    printf("\n정점 %d에서 정점 %d까지의 경로 : ", start, end);
    print_path(path, start, end); // 정점 start에서 정점 end으로 가는 최단경로 출력
    printf("\n");

    return 0;
}
```

실행 예 : (test data1)

Microsoft Visual Studio 디버그 콘솔

	0	1	2	3	4	5	6
=====							
0 :	0	7	*	*	3	10	*
1 :	7	0	4	10	2	6	*
2 :	*	4	0	2	*	*	*
3 :	*	10	2	0	11	9	4
4 :	3	2	*	11	0	*	5
5 :	10	6	*	9	*	0	*
6 :	*	*	*	4	5	*	0
=====							
k=0							
	0	1	2	3	4	5	6
=====							
0 :	0	7	*	*	3	10	*
1 :	7	0	4	10	2	6	*
2 :	*	4	0	2	*	*	*
3 :	*	10	2	0	11	9	4
4 :	3	2	*	11	0	13	5
5 :	10	6	*	9	13	0	*
6 :	*	*	*	4	5	*	0
=====							
k=1							
	0	1	2	3	4	5	6
=====							
0 :	0	7	11	17	3	10	*
1 :	7	0	4	10	2	6	*
2 :	11	4	0	2	6	10	*
3 :	17	10	2	0	11	9	4
4 :	3	2	6	11	0	8	5
5 :	10	6	10	9	8	0	*
6 :	*	*	*	4	5	*	0
=====							
k=2							
	0	1	2	3	4	5	6
=====							
0 :	0	7	11	13	3	10	*
1 :	7	0	4	6	2	6	*
2 :	11	4	0	2	6	10	*
3 :	13	6	2	0	8	9	4
4 :	3	2	6	8	0	8	5
5 :	10	6	10	9	8	0	*
6 :	*	*	*	4	5	*	0
=====							
k=3							
	0	1	2	3	4	5	6
=====							
0 :	0	7	11	13	3	10	17
1 :	7	0	4	6	2	6	10
2 :	11	4	0	2	6	10	6
3 :	13	6	2	0	8	9	4
4 :	3	2	6	8	0	8	5
5 :	10	6	10	9	8	0	13
6 :	17	10	6	4	5	13	0
=====							

k=4							
	0	1	2	3	4	5	6
=====							
0 :	0	5	9	11	3	10	8
1 :	5	0	4	6	2	6	7
2 :	9	4	0	2	6	10	6
3 :	11	6	2	0	8	9	4
4 :	3	2	6	8	0	8	5
5 :	10	6	10	9	8	0	13
6 :	8	7	6	4	5	13	0
=====							
k=5							
	0	1	2	3	4	5	6
=====							
0 :	0	5	9	11	3	10	8
1 :	5	0	4	6	2	6	7
2 :	9	4	0	2	6	10	6
3 :	11	6	2	0	8	9	4
4 :	3	2	6	8	0	8	5
5 :	10	6	10	9	8	0	13
6 :	8	7	6	4	5	13	0
=====							
k=6							
	0	1	2	3	4	5	6
=====							
0 :	0	5	9	11	3	10	8
1 :	5	0	4	6	2	6	7
2 :	9	4	0	2	6	10	6
3 :	11	6	2	0	8	9	4
4 :	3	2	6	8	0	8	5
5 :	10	6	10	9	8	0	13
6 :	8	7	6	4	5	13	0
=====							

정점 0에서 정점 3까지의 경로 : 0->4->1->2->3

실행 예 : (test data2)

선택 Microsoft Visual Studio 디버그

	0	1	2	3	4	5	6
0	0	7	*	5	*	*	*
1	7	0	8	9	7	*	*
2	*	8	0	*	5	*	*
3	5	9	*	0	15	6	*
4	*	7	5	15	0	8	9
5	*	*	*	6	8	0	11
6	*	*	*	*	9	11	0

k=0

	0	1	2	3	4	5	6
0	0	7	*	5	*	*	*
1	7	0	8	9	7	*	*
2	*	8	0	*	5	*	*
3	5	9	*	0	15	6	*
4	*	7	5	15	0	8	9
5	*	*	*	6	8	0	11
6	*	*	*	*	9	11	0

k=1

	0	1	2	3	4	5	6
0	0	7	15	5	14	*	*
1	7	0	8	9	7	*	*
2	15	8	0	17	5	*	*
3	5	9	17	0	15	6	*
4	14	7	5	15	0	8	9
5	*	*	*	6	8	0	11
6	*	*	*	*	9	11	0

k=2

	0	1	2	3	4	5	6
0	0	7	15	5	14	*	*
1	7	0	8	9	7	*	*
2	15	8	0	17	5	*	*
3	5	9	17	0	15	6	*
4	14	7	5	15	0	8	9
5	*	*	*	6	8	0	11
6	*	*	*	*	9	11	0

k=3

	0	1	2	3	4	5	6
0	0	7	15	5	14	11	*
1	7	0	8	9	7	15	*
2	15	8	0	17	5	23	*
3	5	9	17	0	15	6	*
4	14	7	5	15	0	8	9
5	11	15	23	6	8	0	11
6	*	*	*	*	9	11	0

k=4

	0	1	2	3	4	5	6
0	0	7	15	5	14	11	23
1	7	0	8	9	7	15	16
2	15	8	0	17	5	13	14
3	5	9	17	0	15	6	24
4	14	7	5	15	0	8	9
5	11	15	13	6	8	0	11
6	23	16	14	24	9	11	0

k=5

	0	1	2	3	4	5	6
0	0	7	15	5	14	11	22
1	7	0	8	9	7	15	16
2	15	8	0	17	5	13	14
3	5	9	17	0	14	6	17
4	14	7	5	14	0	8	9
5	11	15	13	6	8	0	11
6	22	16	14	17	9	11	0

k=6

	0	1	2	3	4	5	6
0	0	7	15	5	14	11	22
1	7	0	8	9	7	15	16
2	15	8	0	17	5	13	14
3	5	9	17	0	14	6	17
4	14	7	5	14	0	8	9
5	11	15	13	6	8	0	11
6	22	16	14	17	9	11	0

정점 6에서 정점 1까지의 경로 : 6->4->1

문제 3) 방향성 그래프에서 위상 정렬 알고리즘

방향성을 가지는 그래프에서 각 정점들의 선행 순서를 지키면서 모든 정점을 나열하는 그래프의 위상정렬을 수행하는 프로그램을 작성하시오. 위상 정렬의 결과는 여러 가지가 나올 수 있으며, 선행 순서는 반드시 지켜져야 한다. 아래 코드와 실행 예와 같이 결과를 출력하시오.

```
#define TRUE 1
#define FALSE 0
#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;

    graph_init(&g);

    // test data 1
    /*
    set_vertex(&g, 6);
    insert_edge(&g, 0, 2);
    insert_edge(&g, 0, 3);
    insert_edge(&g, 1, 3);
    insert_edge(&g, 1, 4);
    insert_edge(&g, 2, 3);
    insert_edge(&g, 2, 5);
    insert_edge(&g, 3, 5);
    insert_edge(&g, 4, 5);
    */

    // test data 2
    set_vertex(&g, 8);
    insert_edge(&g, 0, 1);
    insert_edge(&g, 1, 2);
    insert_edge(&g, 1, 3);
    insert_edge(&g, 1, 4);
    insert_edge(&g, 2, 7);
    insert_edge(&g, 3, 6);
    insert_edge(&g, 4, 5);
    insert_edge(&g, 4, 6);
    insert_edge(&g, 5, 7);
    insert_edge(&g, 6, 7);

    //위상 정렬
    topo_sort(&g);
    // 동적 메모리 반환 코드 생략
```

<pre> delete_graph(&g); return 0; } </pre>
<p>실행 예 : (test data1)</p> 
<p>실행 예 : (test data2)</p> 