

<알고리즘 과제 2> 코딩과제

문제 1 : 다익스트라(Dijkstra) 최단경로 알고리즘을 활용하여 하단에 출력된 결과를 도출하도록 프로그램을 작성하시오. (C로 작성)

프로그램 요구사항 :

- 1) 가중치 인접행렬(weight adjacency matrix)은 첨부한 알고리즘(code) 참고.
- 2) 최단 경로를 구하는 **shortStep** 함수와 다익스트라 알고리즘을 구현하는 **Dijkstra_Path** 함수는 별도로 작성하는 방향으로 진행.
- 3) 출력예시: 첨부한 결과처럼 단계적으로 실행되도록 구성할 것
- 4) 한 개의 C코드로 작성할 것.

※ **실행참고:** 1) 문자(텍스트) 메시지는 다를 수 있음 (알파벳 권장)

- 2) 숫자(정수)는 첨부된 출력결과와 다른 값으로 실행될 수 있고, 그래프의 명칭(A, B, C 등은 다른 이름으로 설정할 수 있음)

```
***** weight adjacency matrix ***** // 출력메세지는 다를 수 있음
```

```
0  10  5  *  *
```

```
*  0  2  1  *
```

```
*  3  0  9  2
```

```
*  *  *  0  4
```

```
7  *  *  6  0
```

```
***** Dijkstra shortest Path ***** // 출력메세지는 다를 수 있음
```

```
0 step : S={ A }           distance :[ 0  10  5  *  *  ]
```

```
1 step : S={ A  C }        distance :[ 0  8  5  14  7  ]
```

```
2 step : S={ A  C  E }     distance :[ 0  8  5  13  7  ]
```

```
3 step : S={ A  B  C  E }  distance :[ 0  8  5  9  7  ]
```

```
4 step : S={ A  B  C  D  E } distance :[ 0  8  5  9  7  ]
```

가중치 인접행렬(weight adjacency matrix)

```
int weight[MAX_VERTICES][MAX_VERTICES] = {
    { 0, 10, 5, INF, INF },
    { INF, 0, 2, 1, INF },
    { INF, 3, 0, 9, 2 },
    { INF, INF, INF, 0, 4 },
    { 7, INF, INF, 6, 0 },
}; // 정수값은 다를 수 있음

// #define INF 10000           // 무한대 값
// #define MAX_VERTICES n     // 그래프의 정점 개수
```

```
for (i = 0; i < MAX_VERTICES; i++) {
    for (j = 0; j < MAX_VERTICES; j++) {
        if (weight[i][j] == INF)
            printf("%4c", '*');
        else printf("%4d", weight[i][j]);
    }
    printf("\n\n");
}
```

다익스트라 알고리즘

```
Dijkstra_Path(G, v)
    S ← {v};
    for (i ← 0; i ∈ V(G); i ← i+1) do
        distance[i] ← weight[v][i];
    for (i ← 0; S != V(G); i ← i+1) do {
        u ← S에 속하지 않은 정점 중에서 distance[]가 최소인 정점;
        S ← S ∪ {u};
        for (w ← 0; w ∈ V(G); w ← w+1) do
            if (w ≠ S) then
                distance[w] ← min(distance[w], distance[u] + weight[u][w]);
    }
end Dijkstra_Path()
```

```
int shortStep(int step) {
..... for (i = 0; i < MAX_VERTICES; i++)
    if (S[i] == TRUE)
..... if (step < 1) printf(" } WtWtWt");
    else if (step < 4) printf(" } WtWt");
-----

void Dijkstra_Path(int start, int n) {
..... for (i = 0; i < n; i++) {
    distance[i] = weight[start][i];

..... if (distance[u] + weight[u][w] < distance[w])
```

체크 사항 :

- 1) OJ에 제출하지 않음 – 비주얼스튜디오 or 기타 툴킷으로 작성
- 2) i> 작성한 코드 + ii> 실행결과(이미지캡처)를 압축하여 기한내에 블랙보드 “과제2-1” 메뉴에 제출
-> i> 과 ii> 둘 다 제출.

※ **제출기한** : 2022.12.08. - 2022.12.21.(23:59) 까지

문제 2 : 플로이드(floyd) 알고리즘을 활용하여 하단에 출력된 결과를 도출하도록 프로그램을 작성하시오. (C로 작성)

프로그램 요구사항 :

- 1) 가중치 인접행렬(weight adjacency matrix)은 첨부한 알고리즘(code) 참고.
- 2) 최단 경로를 구하는 **shortStep** 함수와 플로이드 알고리즘을 구현하는 **Floyd_Path** 함수는 별도로 작성하는 방향으로 진행.
- 3) 출력예시: 첨부한 결과처럼 단계적으로 실행되도록 구성할 것
- 4) 한 개의 C코드로 작성할 것.

※ **실행참고:** 1) 문자(텍스트) 메시지는 다를 수 있음 (알파벳 권장)
 2) 숫자(정수)는 첨부된 출력결과와 다른 값으로 실행될 수 있고, 그래프의 명칭(A, B, C 등은 다른 이름으로 설정할 수 있음)

```
***** weight adjacency matrix ***** // 출력메세지는 다를 수 있음

0  10  5  *  *

*  0  2  1  *

*  3  0  9  2

*  *  *  0  4

7  *  *  6  0

***** Floyd shortest Path***** // 출력메세지는 다를 수 있음
F-1 :  0  10  5  *  *

      *  0  2  1  *

      *  3  0  9  2

      *  *  *  0  4

      7  *  *  6  0
```

F0 : 0 10 5 * *

 * 0 2 1 *

 * 3 0 9 2

 * * * 0 4

 7 17 12 6 0

F1 : 0 10 5 11 *

 * 0 2 1 *

 * 3 0 4 2

 * * * 0 4

 7 17 12 6 0

F2 : 0 8 5 9 7

 * 0 2 1 4

 * 3 0 4 2

 * * * 0 4

 7 15 12 6 0

F3 : 0 8 5 9 7

 * 0 2 1 4

 * 3 0 4 2

 * * * 0 4

 7 15 12 6 0

```
F4 :    0   8   5   9   7

        11   0   2   1   4

        9   3   0   4   2

        11  19  16   0   4

        7  15  12   6   0
```

// 가중치 인접행렬은 다익스트라 코드 참고

플로이드 최단 경로 알고리즘

```
Floyd_Path(G, v)
  for (k ← 0; k ≤ n-1; k ← k+1) do
    for (v ← 0; v ≤ n-1; v ← v+1) do
      for (w ← 0; w ≤ n-1; w ← w+1) do
        A[v][w] ← min(A[v][w], A[v][k]+A[k][w]);
      end Floyd_Path()
    end Floyd_Path()
  end Floyd_Path()
```

```
int shortStep(int step) {
.....
for (i = 0; i < MAX_VERTICES; i++) {
..... for (j = 0; j < MAX_VERTICES; j++) {
..... if (A[i][j] == INF)
```

```
void Floyd_Path(int n) {
..... for (v = 0; v < n; v++)
..... A[v][w] = weight[v][w];
```

체크 사항 :

- 1) OJ에 제출하지 않음 – 비주얼스튜디오 or 기타 툴킷으로 작성
- 2) i> 작성한 코드 + ii> 실행결과(이미지캡처)를 압축하여 기한내에 블랙보드 “과제2-1” 메뉴에 제출
-> i> 과 ii> 둘 다 제출.

※ **제출기한** : 2022.12.08. - 2022.12.21.(23:59) 까지