**프로그래밍 프로젝트: shortest path finding**

**• 그래프 데이타화일:**

다음 네모와 같은 데이타 화일이 주어진다고 가정한다.

|  |
| --- |
| 12  0 3 8.1 2 2.0 8 2.9  1 4 6.8 7 2.4 2 13.2  2 5 1.3 3 4.9 7 10.8 11 3.2  3 6 5.9 1 5.3 5 3.7  4 10 9.2 11 2.6 2 0.4  5 0 13.1 6 4.2 1 1.5 8 1.1  6 4 3.9 1 7.2  7 4 3.6 9 3.0  8 2 5.1 9 4.1  9 2 1.4 11 7.5  10 7 4.5 9 2.1  11 7 9.1 10 6.0 |

데이타 화일 설명:

- 첫 행은 그래프가 가진 전체 정점수를 나타낸다 (위의 경우 총 12 개의 정점이 있다.)

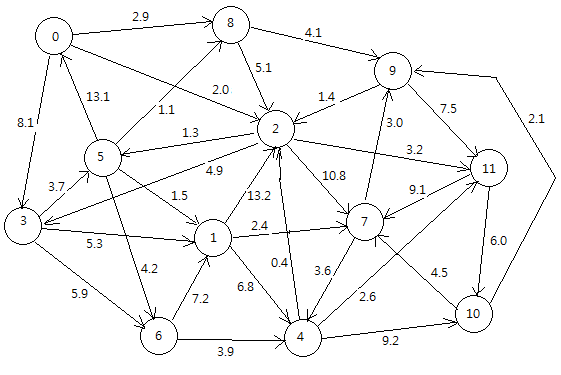
- 둘째 행부터는 다음과 같은 형태를 가진다:

v u1 c1 u2 c2 ...

여기서 v 는 정점이며 그 다음부터는 v에서 나온 아크가 들어오는 정점의 번호, 그 아크의 비용으로 구성된 쌍이 연속적으로 온다. 즉 (v, u1), (v, u2), ... 는 방향성 아크이며, c1, c2, ... 는 해당 아크의 비용이다.

- 이와 같은 행이 정점의 수만큼 온다.

- 예를 들면 위 데이타 화일에 대응하는 그래프의 모양은 다음과 같다:



**•** 프로그램 개발: 다음과 같이 동작하는 프로그램을 개발한다.

작업 1: 그래프를 나타내는 데이타 화일 (화일명: graphdata.txt 로 통일함) 을 읽어서 프로그램 안에 인접행렬 표현인 행렬 cost (각 셀은 1 대신에 해당 아크의 비용을 가짐)를 만든다. 이 작업은 다음과 같은 함수를 호출하여 수행토록 한다.

void Read\_and\_make\_graph ( ) { ....... }

데이타구조 선언은 다음을 이용한다.

#define Max\_vertex 12

double COST [Max\_vetex][ Max\_vetex] ; // 비용정보를 가지는 인접행렬 표현. 전역변수로 선언함.

작업 2: 이 그래프에 대하여 Dijkstra algorithm 을 변경하여 두 정점 사이의 최소경로를 찾아주는 다음 함수를 작성한다.

함수명: int shortest\_path (int start, int destination) { ......}

이 함수를 호출하면 start 에서 출발하여 destination 으로 가는 최소 비용 경로 및 그 경로의 총 비용을 출력한다. 이 함수는 Dijkstra 알고리즘을 기반으로 개발한다. start 를 시작 정점으로 하여 Dijkstra 알고리즘을 수행하는 도중에 destination 정점으로 최소 경로 비용이 결정되자 마자 바로 이 경로 및 그 경로의 총비용을 출력하고 알고리즘을 종료한다.

전역변수로 선언된 distance, pred, set\_S (집합 S) 를 이용하도록 한다:

double distance [Max\_vetex ]; // 알고리즘이 사용하는 거리정보.

int pred [Max\_vetex] ; // 선행자 정보.

int set\_S [Max\_vetex] ; // 각 원소마다 0 은 해당 정점이 S 에 들어있지 않음을, 1 은 들어있음을 나타낸다.

**• 프로그램 수행 예:**

main 함수가 실행되면 먼저 Read\_and\_make\_graph 함수를 호출하여 데이타 화일을 읽어서 COST 행렬을 만든다. 그 다음 다음과 같은 명령 루프를 수행한다:

최소경로를 찾을 두 정점을 입력하시요> 3 9

경로: 3, 7, 6, 5, 9 총비용: 47.3

최소경로를 찾을 두 정점을 입력하시요> 6 1

경로: 6, 7, 6, 5, 1 총비용: 17.3

최소경로를 찾을 두 정점을 입력하시요> 8 3

경로: 8, 7, 6, 5, 3 총비용: 31.8

....

최소경로를 찾을 두 정점을 입력하시요> -1 -1

프로그램을 종료합니다.