/\* 본 프로그램은 최대 10개의 노드를 갖는 그래프의 Prim 알고리즘을 구현한 프로그램임 \*/

#include<stdio.h>

int a, b, u, v, n, i, j, ne = 1;

int visited[10] = { 0 }, min, mincost = 0, cost[10][10];

/\* visited는 노드가 선택되었는지를 확인하는 배열이고 cost가 가중치 그래프를 나타내는 배열임 \*/

void main()

{

printf("\nEnter the number of nodes:");

scanf("%d", &n);

printf("\nEnter the adjacency matrix:\n");

for (i = 1; i <= n; i++)

for (j = 1; j <= n; j++)

{

scanf("%d", &cost[i][j]);

if (cost[i][j] == 0) // 링크가 없을 경우

cost[i][j] = 999; //최소값 링크 선택을 위해 큰수 배정

}

visited[1] = 1; // 루트를 1로 지정 또한 1은 트리에 선택되었음을 나타냄

printf("\n");

while (ne < n) // ne는 트리에 선택된 링크 수를 나타내는 변수임 초기값은 1임

{

for (i = 1, min = 999; i <= n; i++) // 모든 노드 i 각각에 대해

for (j = 1; j <= n; j++) // i 와 연결된 노드 모든 j에 대하여

if (/\*링크 (I, j)의 값이 min보다 작을 경우\*/)

if (/\*노드 i가 트리에 선택되었을 경우\*/)

{

min = cost[i][j];

a = i;

b = j;

}

if (visited[a] == 0 || visited[b] == 0)

{

printf("\n Edge %d:(%d %d) cost:%d", ne++, a, b, min);

mincost += min;

visited[b] = 1;

}

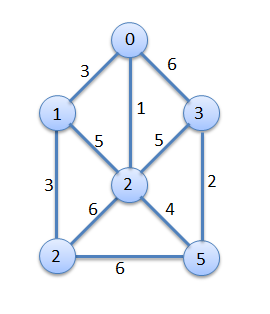
cost[a][b] = cost[b][a] = 999;

}

printf("\n Minimun cost=%d", mincost);

getch();

}



실행

Enter the number of nodes : 6

Enter the adjacency matrix:

0 3 1 6 0 0

3 0 5 0 3 0

1 5 0 5 6 4

6 0 5 0 0 2

0 3 6 0 0 6

0 0 4 2 6 0

Edge 1: (1 3) cost: 1

Edge 2: (1 2) cost: 3

Edge 3: (2 5) cost: 3

Edge 4: (3 6) cost: 4

Edge 5: (6 4) cost: 2

Minimum cost = 13