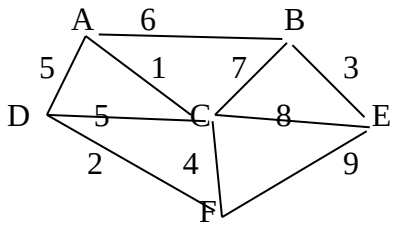
Homework 7 Kruskal's Algorithm Due: 6/10

1. 다음 그래프를 데이터로 이용할 것 (Adjacency List로 저장)



A 6 B B 3 E
B 7 C C 8 E
A 1 C C 4 F
A 5 D D 2 F
C 5 D E 9 F

2. 실행 결과 (Output)

Input Data:

A 6 B, B 7 C, A 1 C, A 5 D, C 5 D

B 3 E, C 8 E, C 4 F, D 2 F, E 9 F

Sorted Data:

A 1 C, D 2 F, B 3 E, C 4 F, A 5 D

C 5 D, A 6 B, B 7 C, C 8 E, E 9 F

Kruskal's MST:

1) Edge1: A 1 C
2) Edge2: D 2 F
3) Edge3: B 3 E
4) Edge4: C 4 F
5) Edge5: A 6 B.

Total Cost: 16

3. 알고리즘

- ADT (kruskal, check_cycle, sort.)
  Check_cycle: you can use your own algorithm
- Sort : Use any sorting algorithm

- Vertex 정의:

```
struct inputdata {    char ff;    int edges; char ll;}v[100];

struct list_node {    char vertex;    struct list_node *link; } *table[10];
```

Kruskal's Algorithm:

1) Print Input data

2) Print Sorted Data

3) Make head node for all (from A to F) into Array (DFS의 인접리스트 생성처럼 – 다수의 single list)

4) for (all the sorted edges) {

check cycle for each edge (ex A B)

if (!=cycle) then print MST edge, sum++ for total weight;

}

Check_cycle(NodeA NodeB) {

Find NodeA location from Head Node Table & make List1

- Move to the end of List1 if possible

Find NodeB location from Head Node Table & make List2

- Move to the end of List2 if possible

If (List1->vertex == List2->vertex) return cycle, else no-cycle

}

4. 실행 화면