# # Modify FIND

Procedure FIND (i)

Description - Find the root of tree containing element i use the collapsing rule for collapsed.

Decloration -

integer i, j, k

Algorithm -

```
j ← i
// first find a root of three
    while PARENT (j) > 0, do
        j ← PARENT (j)
    repeat
    return (j)
// collapse the nodes from node (i)
   up to the root (j)
    k ← i
    while k ≠ j, do
        temp ← PARENT (k)
        PARENT (k) ← j
        k ← ~~PA~~ temp
    repeat
    return (j)
```

END_FIND

Remarks ______________________________ Returned on ______________

# Modify UNION

Procedure UNION (i,j)

Description - UNION sets with roots i and j, i ≠ j, using the weighting rule.

```
    PARENT (i) ← - COUNT (i) and
    PARENT (j) ← - COUNT (j)
Declaration -
        integer (i), j, x
Algorithm -
        x ← PARENT (i) + PARENT(j)
        if PARENT (i) > PARENT (j), then
            PARENT (i) ← j.
            PARENT(j) ← x
        else
            PARENT (j) ← i
            PARENT (i) ← x
        endif
END UNION
```

complete for :
Algorithm
Flow Chart
Programme Listing
Results
Comments

Remarks ______ Submitted on ______

Returned on ______

# Simple Union Algorithm :-

```
Procedure U(i,j)
    // Description :-
        Replace the disjoint sets with roots
        (i,j), i ≠ j by their union
    // Declaration :-
        integer i,j
    // Algorithm
        PARENT(i) ← j
END U
```

# Simple Find Algorithm

```
Procedure F(i)
    // Description :-
        Find the root of the tree containing element i.
    // Declaration :-
        integer i,j;
    // Algorithm :-
        j ← i
        while PARENT(j) > 0
            j ← PARENT(j)
        repeat
        return (j)
END F.
```

omplete for :

lgorithm

Flow Chart

Programme Listing

Results

Comments