

## Disjoint Sets (20 pts)

### Problem Description

In this problem, we ask you to implement the standard disjoint-set forest data structure and algorithm presented in the lecture and in the textbook. The pseudo codes of the three main operations of the disjoint set data structure, MAKE-SET, UNION, and FIND-SET, are as follows. The pseudo code implements the heuristics to improve the running time: *union by size* and *path compression*. You also need to implement a PRINT-PATH function, which prints all nodes of the path from the given node to the root of the disjoint-set tree that has the given node, following the parent link of each visited node.

Note that you need to follow exactly how to union two sets (by their sizes) and how to perform path compression, in order to have PRINT-PATH generate the right answer.

Assume the object  $x$  which forms the disjoint sets are represented by distinct positive integers,  $1 \leq x \leq N$ . Printing an object  $x$  would output that integer. In addition, assume that MAKE-SET is executed for each of the  $N$  objects (i.e., for these  $N$  integers) as part of the initialization process.

MAKE-SET( $x$ )

```
1   $x.p = x$ 
2   $x.rank = 0$ 
```

UNION( $x, y$ )

```
1   $v = \text{FIND-SET}(x)$ 
2   $w = \text{FIND-SET}(y)$ 
3  if  $v \neq w$ 
4      LINK( $v, w$ )
```

FIND-SET( $x$ )

```
1  if  $x \neq x.p$ 
2       $x.p = \text{FIND-SET}(x.p)$ 
3  return  $x.p$ 
```

```

LINK( $x, y$ )
1  if  $x.rank > y.rank$ 
2       $y.p = x$ 
3  else
4       $x.p = y$ 
5      if  $x.rank == y.rank$ 
6           $y.rank = y.rank + 1$ 

```

```

PRINT-PATH( $x$ )

```

```

1   $k = x$ 
2  while  $k.p \neq k$ 
3      print  $k$ 
4       $k = k.p$ 
5  print  $k$ 

```

## Input

The first line includes  $N$ . The second line includes  $M$ , which specifies the number of operations in the input. In each of the following  $M$  lines, it will be in one of these three formats:

1. **F**  $x$ . This means you should execute  $\text{FIND-SET}(x)$ .
2. **U**  $x$   $y$ . This means you should execute  $\text{UNION}(x, y)$ .
3. **P**  $x$ . This means you should execute  $\text{PRINT-PATH}(x)$ .

The  $x$  and  $y$  above are integers representing the sets. A single white space is used to separated numbers and after the characters **F**, **U**, and **P**.

You should execute all operations according to the given order.

## Output

For each **P** operation (i.e.,  $\text{PRINT-PATH}$ ) given, output in one line, the output of  $\text{PRINT-PATH}$ , which include all the nodes in the path from the given nodes  $x$  to the root of the disjoint-set tree that  $x$  belongs to, in this format:  $x \ x_1 \ x_2 \ \dots \ x_n$  Consecutive numbers are separated by a white space.

## Constraint

- $1 \leq N \leq 1000000$
- $1 \leq M \leq 1000000$

## Sample Testcases

### Sample Input 1

3  
6  
F 1  
U 3 1  
U 3 3  
U 2 2  
F 2  
P 3

### Sample Output 1

3 1

### Sample Input 2

5  
10  
U 4 5  
F 5  
P 4  
P 5  
P 3  
U 5 4  
U 2 4  
F 2  
U 2 5  
P 2

### Sample Output 2

4 5  
5  
3  
2 5

### Sample Input 3

5  
10  
U 3 4  
U 1 4  
F 4  
P 1  
P 4  
P 3  
F 1  
P 2  
P 1  
P 4

### Sample Output 3

1 4  
4  
3 4  
2  
1 4  
4