Programming Assignment 05
Submit on 18 April 2019

Problem statement

In this program you will implement Union-Find data structure using a linked list as described in the class. The data structure will store any partition of the set of integers $\{0, 1, 2, \ldots, n-1\}$. Each set will be stored in one linked list. There will be one linked-list node (called *ListNode*) for each integer. Each node (called *ListNode*) in the linked list will have three fields: *element*, *set*, *next* storing the integer, set identity, and the next pointer respectively. The *ListNode* for i will permanently store i in field *element*. The *set* field will store the identity of the set to which this element currently belongs. So this value may change when set union operation is performed.

In addition there will be three arrays, called *Element*, *Set*, and *Size*, each of size n. $Element[i]$ will be a pointer pointing to the *ListNode* of integer i. $Set[j]$ will point to the first node of the linked list of the set j. $Size[j]$ will be store the number of elements in set j.

Initially all sets will be singletons. For each $i \in \{0, 1, \ldots, n-1\}$, set i will contain integer i initially. So at the start $Set[i]$ will point to the node of integer i and this linked list will have no other node. Also initially $Size[i] = 1$ for each i. Similarly *set* field of *ListNode* of i will store i initially.

Observe that the set identity of integer i is $Element[i].set$ at any time. So the set identity cn be found in $O(1)$ time. To perform the union of sets j and k is performed as follows. (i) First compare $Size[j]$ and $[Size[k]$. Suppose $Size[j] \leq Size[k]$. Go through each *ListNode* of the linked list pointed by $Set[j]$ and store k in their *set* fields. Make the list node to point to the first node of the list pointed by $Set[k]$. Then make $Set[k]$ point to the first node of the combined linked list by doing $Set[k] := Set[j]$ and set $Set[j] = null$. The update $Size[j]$ and $Size[k]$.

Operations

Implement the data structure to perform the following operations.

$Union(a, b)$ Perform the union of the sets containing integers a and b and return 1. If these integers already belong to the same set, then return 0.

$Find(a)$ Determine the set identity of the integer a and return the same.

$Same(a, b)$ Return 1 of integers a and b belong to the same set. Otherwise return 0.

Code-instructions

The input will be a sequence of operations which must be performed in order. The operations will be coded as follows.

- **1** *a b*: means $Union(a, b)$.

- **2** *a* : means $Find(a)$.

- **3** *a b*: means $Same(a, b)$.

Input Format

- The first line of the input contains an integer n denoting number of integers in the set and an integer m denoting number of operations.

- In the next m lines there will be one coded operation per line.

Output

Print the output of each instruction on a new line.

Example

Input	Output
4 6	
1 1 3	1
2 2	2
1 2 4	1
3 1 3	1
1 3 1	0
3 2 3	0