**Disjoint set (Union-Find)**

**Easy**Accuracy: 37.74% Submissions: 5868 Points: 2

Given an array **A[]** that stores all number from **1** to **N** (both inclusive and sorted) and **K** queries.

The task is to do the following operations on array elements :

1. **UNION** X Z : Perform union of **X** and **Z** i.e. parent of **Z** will become the parent of **X**.  
2. **FIND** X: Find the parent of **X** and print it.

Note: Initially all are the parent of themselves.

**Input:**

N = 5, K = 4

queries[] = {{find 4},

  {find 1},

  {unionSet 3 1},

  {find 3}}

**Output:**

4 1 1

**Explanation:**

1. The parent of 4 is 4. Hence the output is 4.

2. The parent of 1 is 1. Hence the output is 1.

3. After performing unionSet 3 1, parent of 3 becomes 1,

  since, parent of 1 is currently 1 itself.

4. The parent of 3 is now 1. Hence, the output is 1.

**Your Task:**    
You don't need to read input or print anything. Your task is to complete the functions- **find()** which takes an array **A**[]and an integer **X**as an input parameter and return the parent of **X**and the function **unionSet()**which takes an array **A**[]and two integers **X**and **Z**and performs the **union** of **X** and **Z**.

**Expected Time Complexity:** O(N)  
**Expected Auxiliary Space:** O(1)

**Constraints:**  
1 <= N, K <= 100

int find(int A[],int X) {

    //add code here

    if (A[X]!=X) A[X]=find(A, A[X]);

    return A[X];

}

void unionSet(int A[], int X, int Z) {

    //add code here.

    int rootx=find(A, X);

    int rootz=find(A, Z);

    A[rootx]=rootz;

}