**EX - 5: Disjoint Sets**

**Program Description:**

Disjoint sets are also known as **disjoint collections**or**non-intersecting sets** and refer to those sets which don’t have any identical element common between them.

**Example,** A = {1, 2, a} and B = {3, 4, b} then, A ∩ B = ϕ.

For this example, A and B are disjoint sets as both do not contain the same element.

The disjoint sets are also called **mutually exclusive sets**or**independent sets**.

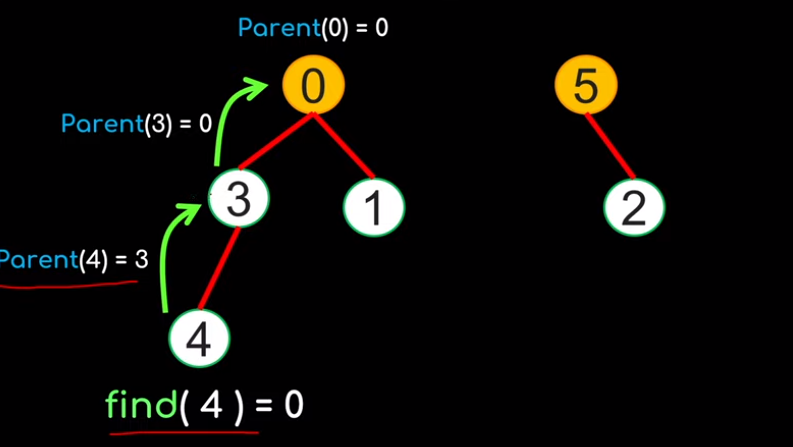
A [disjoint-set data structure](https://www.geeksforgeeks.org/introduction-to-disjoint-set-data-structure-or-union-find-algorithm/) is defined as one that keeps track of a set of elements partitioned into a number of disjoint (non-overlapping) subsets.

The disjoint set data structure supports following operations:

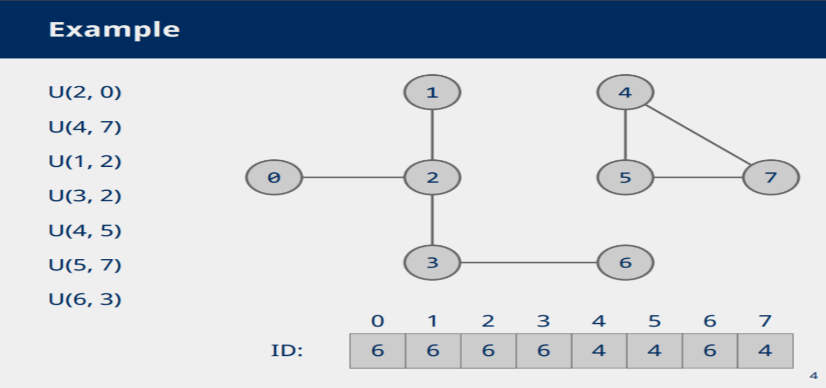
* Adding new sets to the disjoint set.
* Merging disjoint sets to a single disjoint set using **Union**operation.
* Finding representative of a disjoint set using **Find** operation.
* Check if two sets are disjoint or not.

A [union-find algorithm](https://www.geeksforgeeks.org/introduction-to-disjoint-set-data-structure-or-union-find-algorithm/) is an algorithm that performs two useful operations on such a data structure:

* **Find:** Determine which subset a particular element is in. This can determine if two elements are in the same subset.



* **Union:** Join two subsets into a single subset. Here first we have to check if the two subsets belong to the same set. If not, then we cannot perform union.



**Simple Union-Find algorithm:**

Procedure:

1. Initialize every element as considered as separate tree
2. P[i], easch and every node has to maintain parent node address.

P[i] – Parent of Node[i]

3. Root node P[i]=-1

4. Perform Union and update P[i] value

**Algorithm SimpleUnion(I,j)**

{

P[j]=I;

}

**Algorithm SimpleFind(i)**

{

while(P[i]>=0)

{

i=P[i];

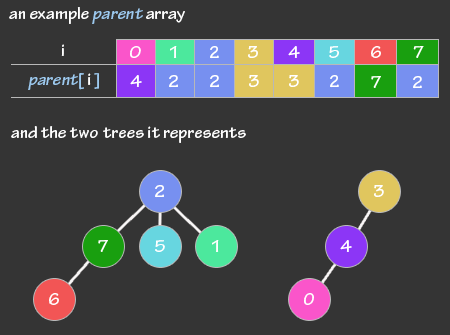
}

return i;

}

**Time complexity**: O(n)

Example:



Applications:

* Union-Find is used to determine the connected components in a graph.
* It is used to determine the cycles in the graph. In the **Kruskal’s Algorithm**, Union Find Data Structure is used as a subroutine to find the cycles in the graph, which helps in finding the minimum spanning tree

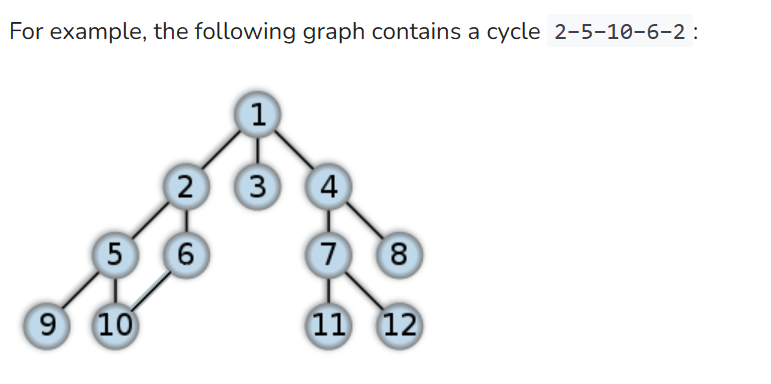
Program code:

1. Implement simple find and union algorithm

Input and Output:

Application programs:

1. <https://www.hackerrank.com/challenges/merging-communities/problem?isFullScreen=true>
2. <https://www.hackerrank.com/challenges/components-in-graph/problem?isFullScreen=true>
3. <https://www.hackerrank.com/challenges/kundu-and-tree/problem?isFullScreen=true>
4. [**https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/practice-problems/algorithm/containers-of-choclates-1/**](https://www.hackerearth.com/practice/algorithms/graphs/breadth-first-search/practice-problems/algorithm/containers-of-choclates-1/)
5. **Given a connected undirected graph, check if it contains any cycle or not.**

****