Graph: Disjoint-Set-Union

# Graph representation

1. Adj matrix
   1. Directed
   2. Weighted
2. Adj list
   1. Directed
   2. Weighted

Practice problems:

1. Print all out-going edges of a given vertex
2. Print the degree of each vertex for an undirected graph
3. Find the min degree of an undirected graph

|  |  |
| --- | --- |
|  |  |

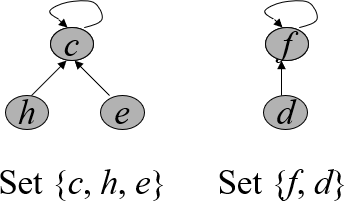
# Disjoint-set

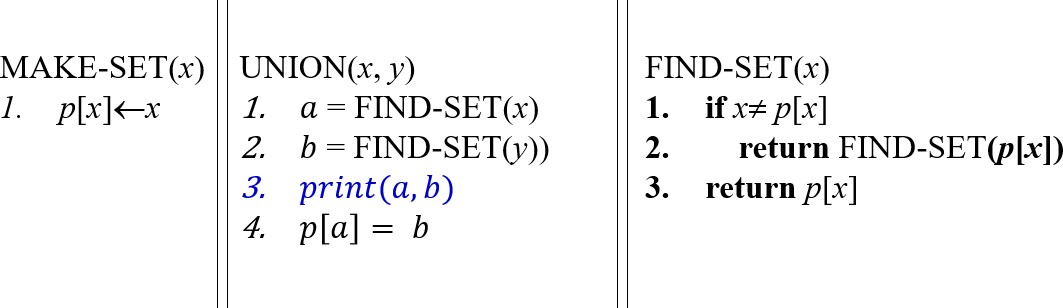
* Make-Set(x) – Creates a new set {x} where x is it’s only element (and therefore it is the

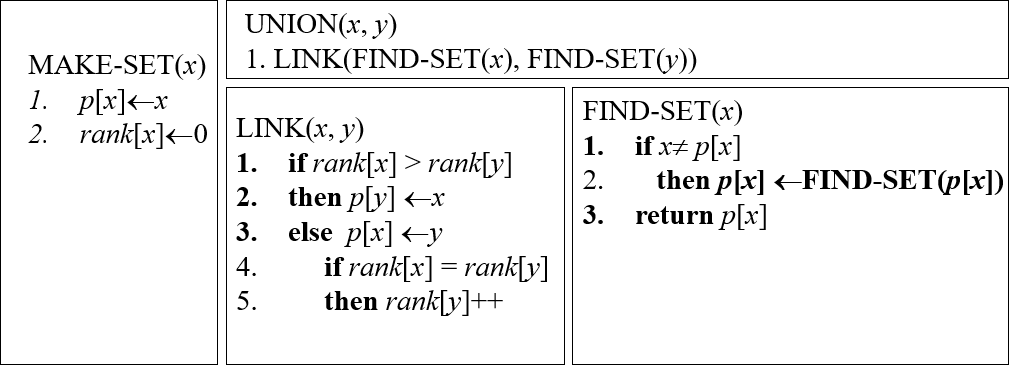
representative of the set).

* Union(x, y) – Merges the set where x belongs with the set where y belongs. One of the elements of the merged set becomes the representative.
* Find(x) – Returns the representative of the set containing x.

## Rooted tree implementation (plain)

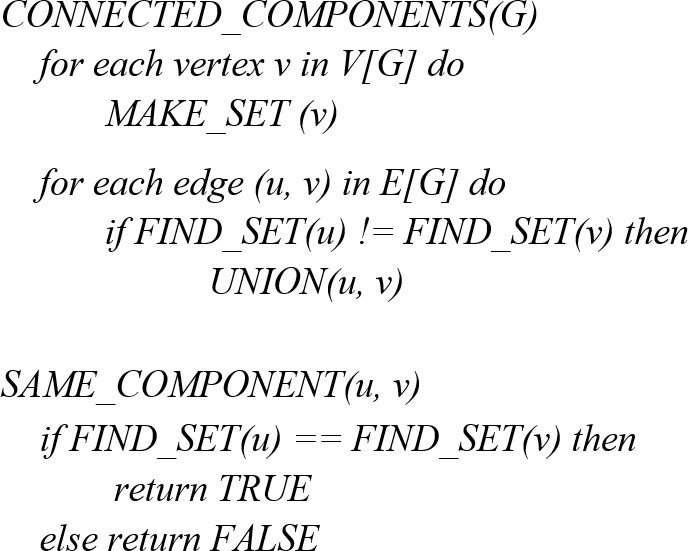




Rooted tree implementation (with union-by-rank and path compression heuristic):

## Practice problems:

1. Given a graph, find how many connected components are there? Print each connected-component. [Hint: use disjoint set data structure]



1. Describe a data structure that supports the following operations:
   1. find(x) – returns the representative of x
   2. union(x, y) – unifies the groups of x and y
   3. min(x) – returns the minimal element in the group of x