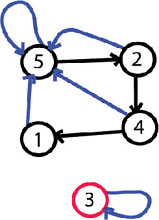
**Disjoint Set**

Operations - MAKE-SET(), FIND-SET(), UNION()

Implementation - Using Linked Lists -> Represent each set with a linked list such that each element in the list points to the next element in the list and also to the representative element.

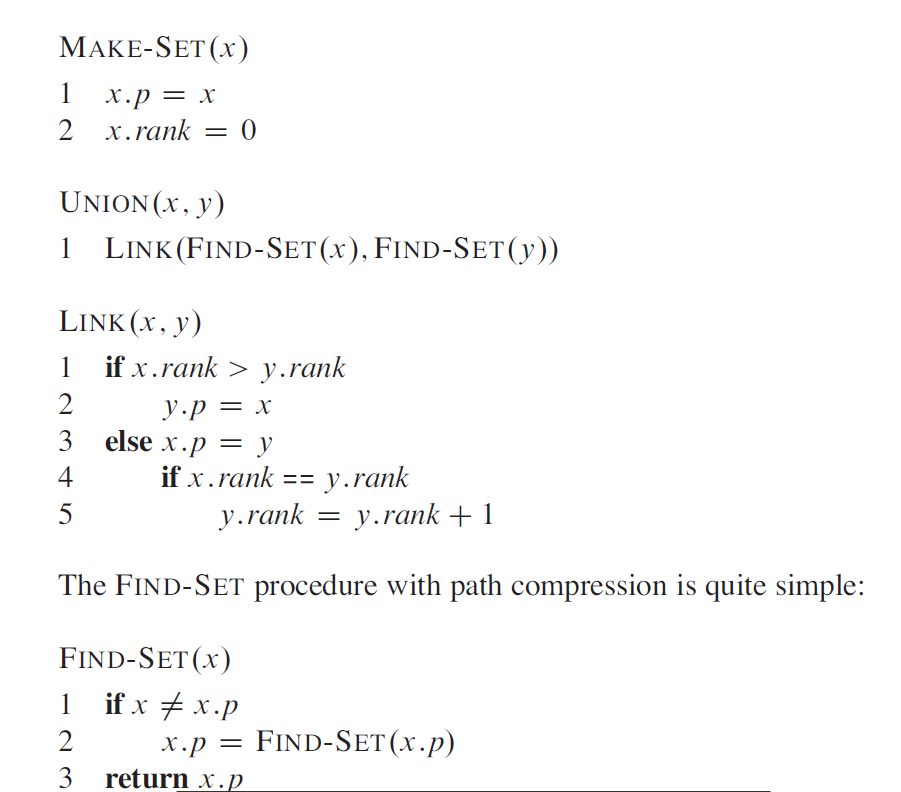
(Choose largest element as representative)



Naive Union - O(n2)

Weighted Median Heuristic - O(nlogn)

Sudo Code -

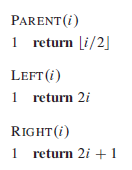


**Heaps -**

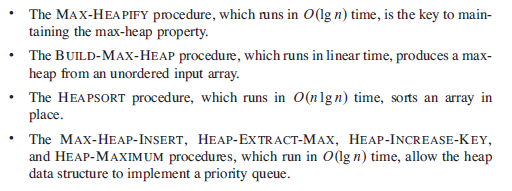
A - Array representing heap.

Max Heap Property - A[Parent] >= A[i] (i - childs)

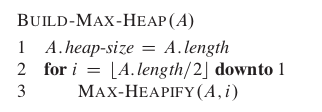
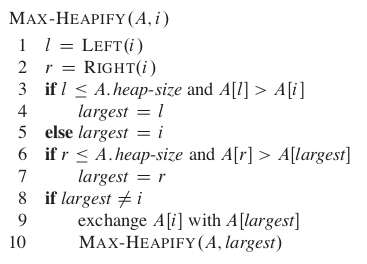
Min Heap Property - A[Parent] <= A[i] (i - childs)

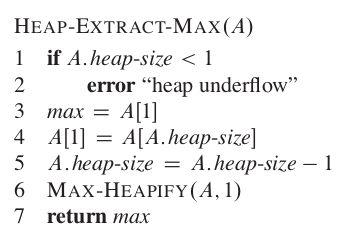
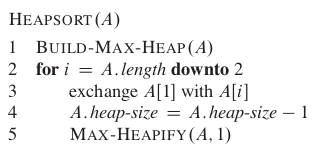


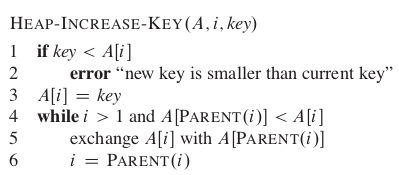
The binary tree is always complete that is apart from the leaves each node have exactly two childs.



Build\_max\_heap is O(n) (upper bound - O(nlogn) but the height of the tree is variable in each call of the max\_heapify.)

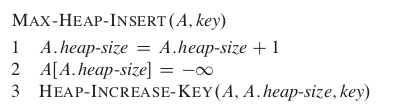






The running time of HEAP - INCREASE -KEY on an n-element heap is O(logn), since the path

traced from the node updated in line 3 to the root has length O(logn).



O(logn) removal from heap(not exactly ;)) - check - Greedy\_ANUMLA.py