**BASICS ON SETS**

Two non-empty sets X and Y are disjoint if their intersection is empty, X ∩ Y = .

**Operations on sets**

makeset(i): constructs set {i}.

Ex: makeset(1), makeset(2), makeset(3) produces the sets {1}, {2}, and {3}.

findset(i): returns the “marked” member of the set to which *i* belongs.

Ex: Given the set {a, b}, findset(b) will return either *a* or *b* depending on which one is the member marked. Some thing for findset(a).

Ex: Given the set {1,4} with *4* being the marked member then findset(1) = findset(4) = 4.

union(i, j): replaces the sets containing *i* and *j* with their union.

Ex: Given {a, b, c} and {1, 2}, then union(b, 1) returns {a, b, c, 1, 2}.

Ex: Assume that in the previous case *b* was the marked element, then some representations for the union set may be:

**b**

**b**

**a**

**c**

**c**

**1**

**1**

**2**

**2**

**2**

**a**

**b**

**a**

**c**

**1**

**2**

**b**

**a**

**c**

**1**

**. . .**

“marked”

* the boolean-valued field mark [x] indicates whether node

*x* has lost a child since last time x was made the child of another node

* newly created nodes are unmarked and node *x* becomes unmarked whenever it is made the child of another node.

Representation of disjoint sets using the array “parent”.

The representation is straightforward.

Ex: Given {a ,b, c, d}, {e} and {30, 60, 17}, assume that yhey are represented by the following trees

**c**

**a**

**d**

**b**

**17**

**30**

**60**

**e (the root)**

Then their representation will be

e

e

b

a

c

c

c

a

a

d

30

60

17

17

17

17

The mergetree operation:

This is also a well known operation. Let’s see it with an example:

**b**

**c**

**d**

**e**

**p**

**q**

**r**

**+**

**a**

mergetree(a, p)

**b**

**c**

**d**

**e**

**a**

**p**

**q**

**r**

yet another example would be:

**a**

**b**

**d**

**c**

**+**

mergetree(a, p)

**p**

**q**

**u**

**r**

**a**

**b**

**d**

**c**

**p**

**q**

**u**

**r**

**p**

**q**

**u**

**r**

**a**

**b**

**d**

**c**

**or**

A mergetree algorithm

mergetree(i, j)

1. nother example would be:tion.ion.60, 17}, assume that yhey are represented by the following treesnode.if(height[i] < height[j] {
2. parent[i] parent[i] = j
3. elselelseelse if(height[i] > height[j]
4. parent[j] = i
5. else {
6. parent[i] = j
7. height[j] = height[j] + 1}
8. }

Examples are shown above, both for trees with the same height and trees os different heights.