

### Recitation Worksheet: Disjoint Sets, 2-4 Trees

1) Show the state of the array storing a disjoint set AFTER it had undergone the following operations. (Note: Assume that the items in the disjoint set of size  $n$  are 0 through  $n-1$ .) Assume that the shorter tree is always attached to the longer tree and that if two trees of equal height are put together that the tree with the higher root value is attached to the tree with the lower root value.

```
DisjointSet dj = new DisjointSet(10);  
dj.union(3, 8);  
dj.union(2, 4);  
dj.union(8, 7);  
dj.union(4, 7);  
dj.union(9, 1);  
dj.union(9, 6);  
dj.union(3, 0);  
dj.union(5, 9);
```

Index	0	1	2	3	4	5	6	7	8	9
Value										

2) Draw the disjoint set (in the form of a forest) stored in the following array:

3	7	6	6	10	6	5	5	5	10
1	2	3	4	5	6	7	8	9	10

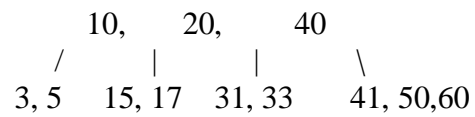
Give the state of the array after executing `union(6,10)`:

1	2	3	4	5	6	7	8	9	10

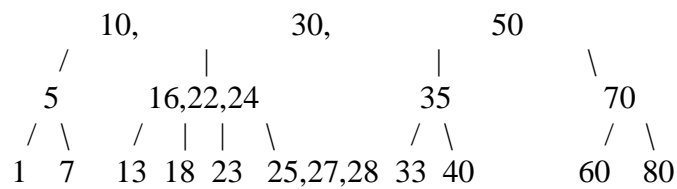
Now go back to the array we had before the union operation. Give the state of the array after executing `findset(2)`. Assume we are using path compression.

1	2	3	4	5	6	7	8	9	10

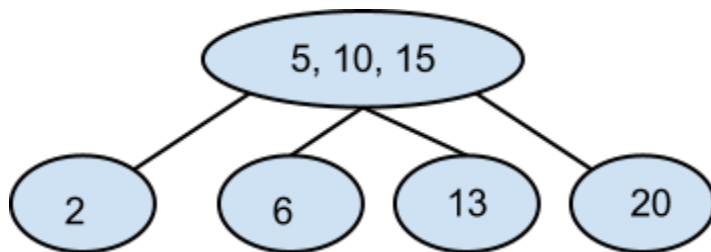
3) Show the final result of inserting the value 47 into the 2-4 tree depicted below.



4) Show the result of inserting 26 into the 2-4 Tree below. Note: Whenever a node overflows, please "send up" the **third** value out of four in the overflowed node.

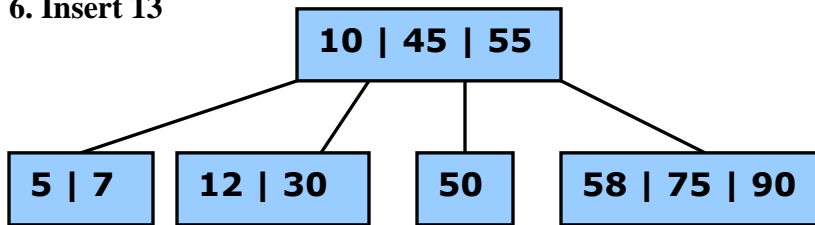


5) Delete 20 from the following 2-4 tree:

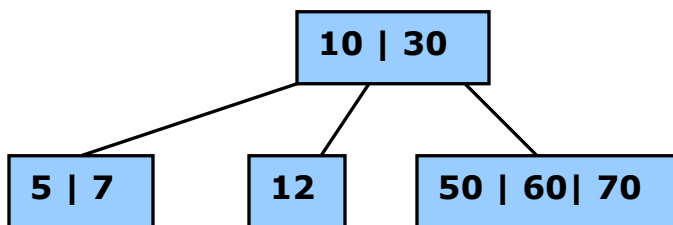


## 2-4 Trees

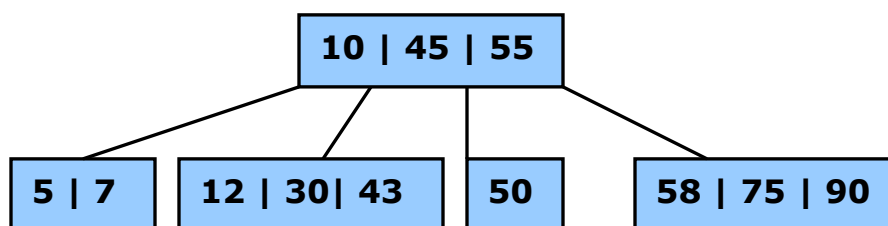
6. Insert 13



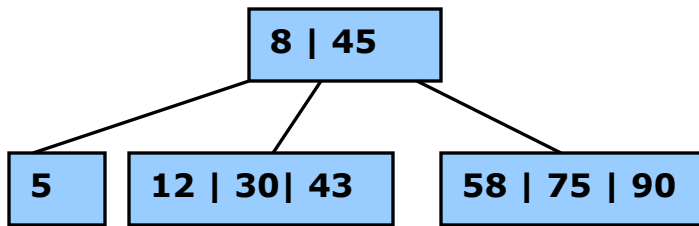
7. Insert 40



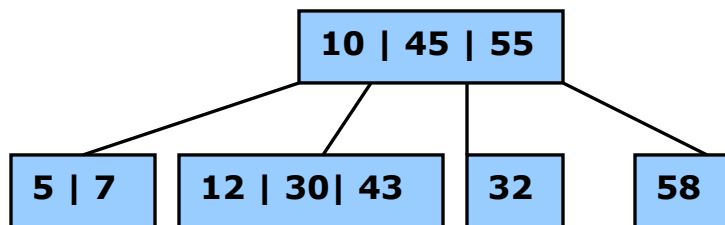
8. Insert 40



9. Delete 5



10. Delete 32



11. Delete 50

