Name: ____________________________________

Email address: ____________________________________

1) **Disjoint Sets:** Consider the set of initially unrelated elements 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14.

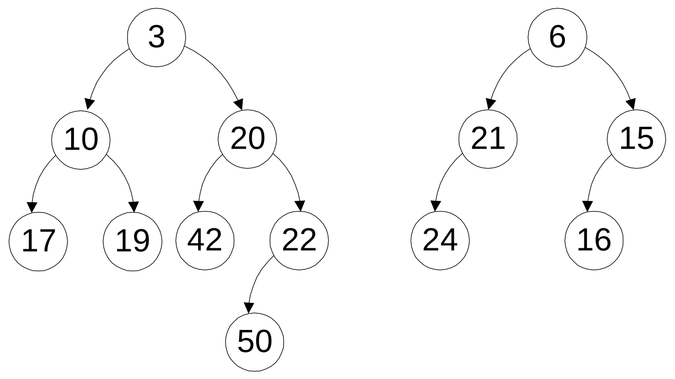
a.) Draw the final forest of up-trees that results from the following sequence of operations using on union-by-size. Break ties by keeping the first argument as the root. Perform a find (without path compression) if you need to find which set an element belongs to.

Union(0,3), Union(3,4), Union(7,9), Union(9,3), Union(7, 14), Union (6,8), Union(6,0), Union(12,6), Union(1,12), Union(11,5), Union(7,11).

b.) Draw the array representation of your answer above. Use the value -1 to represent a root.

c.) Draw the new forest of up-trees that results from doing a Find(9) with path compression on your forest of up-trees from (a).

2) **Leftist Heaps:**
Below are two leftist heaps. Show the result of merging them together using the algorithm we discussed in class. You are only required to show the final tree, although if you draw intermediate trees, ***please circle your final result for ANY credit.*** You may continue your answer to this question on the next page if needed.



3) **Skew Heaps:** Show the result of inserting values in this order into an initially empty skew heap: (1, 2, 6, 3, 4, 8, 7, 5, 9, 10, 11)

4) Draw the contents of the hash table in the boxes below given the following conditions:

The size of the hash table is 8.
Open addressing and quadratic probing is used to resolve collisions.
The hash function used is H(k) = k mod 8

What values will be in the hash table after the following sequence of insertions? Draw the values in the boxes below, and show your work for partial credit.

33, 10, 9, 13, 12

0	
1	
2	
3	
4	
5	
6	
7	

5) Memory

a) Define spatial locality.

b) Give an example of a data structure and a way of accessing that data structure that will have *spatial* locality. Give an example (using code) and indicate **<u>what</u>** will have spatial locality and why.

c) Define temporal locality.

d) Give an example of a data structure and a way of accessing that data structure that has *temporal* locality. Give an example and indicate what will have temporal locality and why.

6) B-trees

Given the following parameters:

Disk access time = 1milli-sec per byte

1 Page on disk = 2048 bytes

Key = 20 bytes

Pointer = 4 bytes

Data = 256 bytes per record (includes key)

What are the best values for: (**Show your work for full credit.**)

M =

L =

7) B-trees: Insert the following values ***in this order*** into a B-tree with L=2 and M=3: (5, 4, 7, 3, 2, 1, 9, 8, 6)