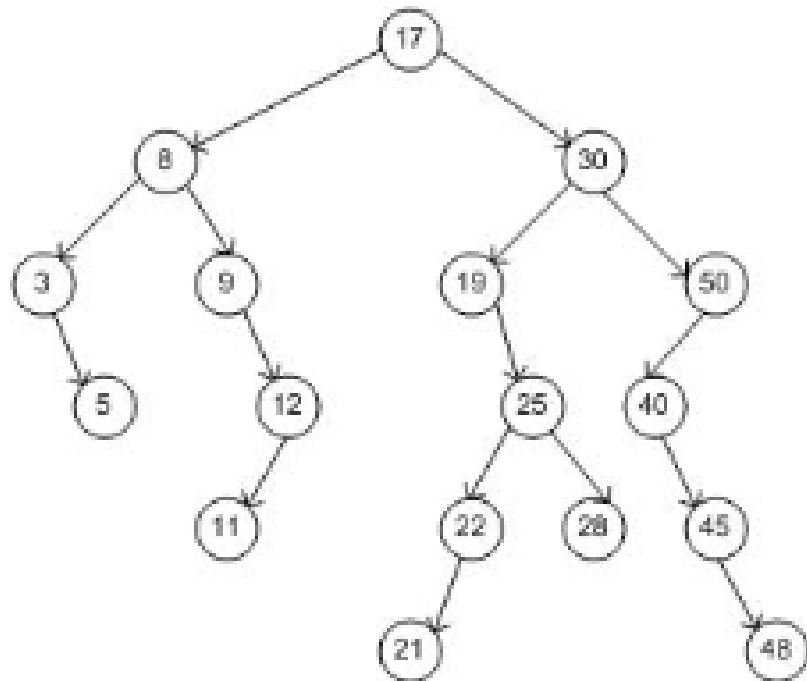
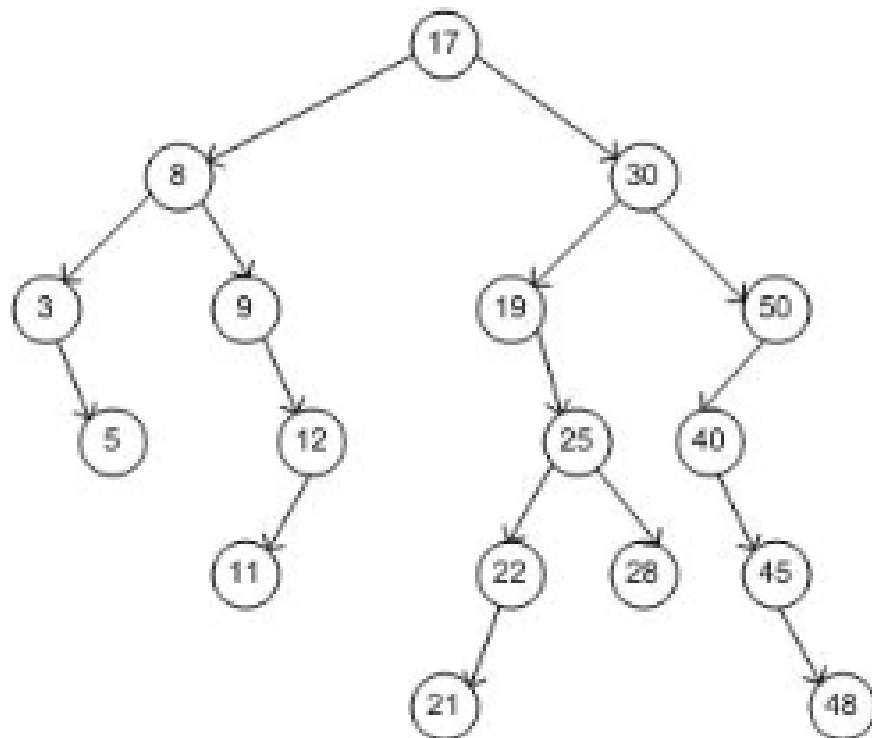


CS 2420 – Written 3
10 Points

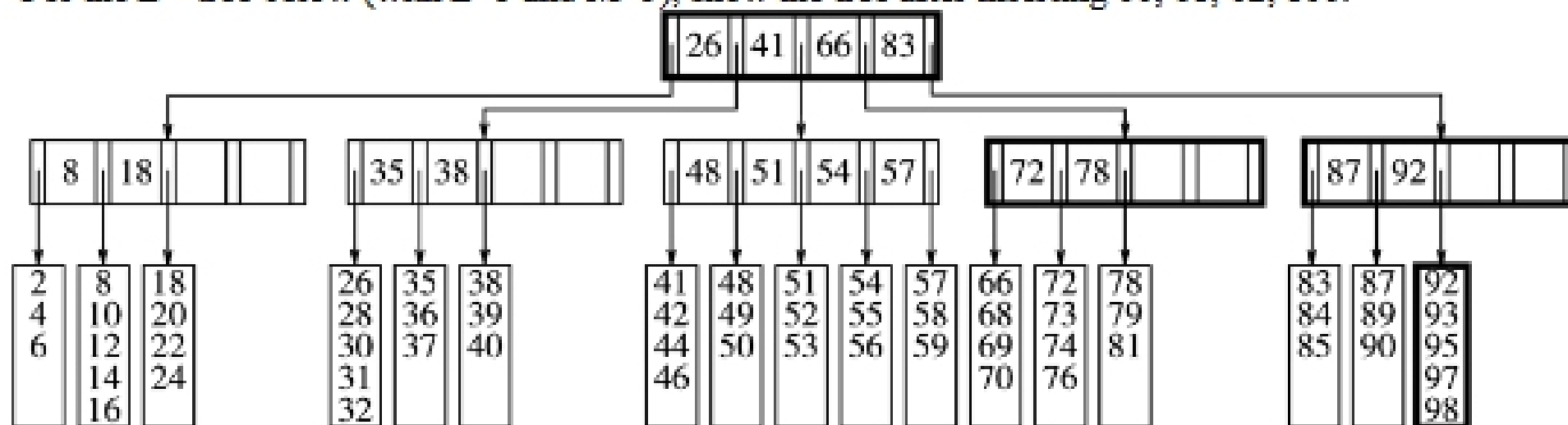
1. Show the result of inserting 10 into the top-down splay tree shown below. (HINT: You can test your answer by going to CS3/Splay Trees. You may find it difficult to create specific tree, as any operation changes the existing tree. Try the following to create a specific tree. **Delete All** and then click **Insert Binary Tree** for each node to insert nodes by level (and not have splaying done). Once you get the tree build, **Insert** will insert using splay tree operations.)



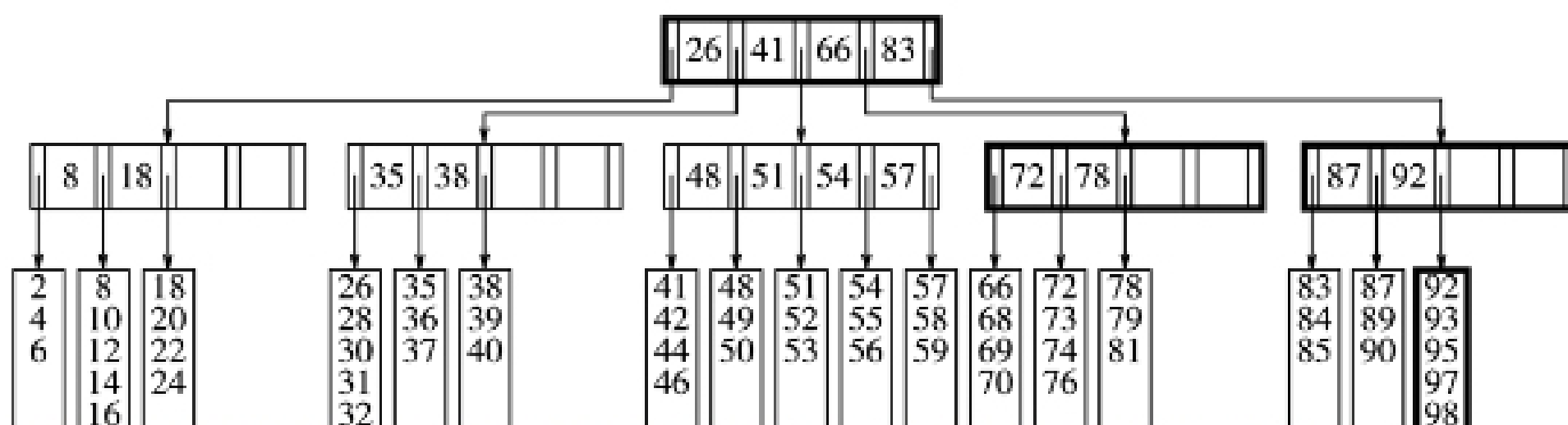
2. Show the result of deleteMax from the top-down splay tree show below:



3. For the B+-tree below (with $L=5$ and $M=5$), show the tree after inserting 60, 61, 62, 100.



4. For the B+-tree below (with $L=5$ and $M=5$), show the tree after deleting 98, 85, 83, 49, 50.



5. The definitions of primary and secondary clustering are sometimes difficult to get straight.

In primary clustering, not only do items collide because they hash to the same location, but one item may collide with the alternate location for another. This occurs in linear probing. When the item that hashes to x is placed in y , items that hash to either x or y both try the same next location. As an analogy, suppose I have one candy bar of each type and am trying to give everyone their first choice. Suppose two people want a Babe Ruth, obviously someone will have to go with second choice. But what if the second person preferring a Babe Ruth got a KitKat (which was someone else's favorite candy bar). So the person who wanted a KitKat has to take a Milky Way. Our concern is that we may be causing more people to be unhappy by causing a longer chain of substitutions that may be necessary.

In secondary clustering, two items that hash to the same location follow the same probe sequence for subsequent locations. This happens in quadratic probing. Only items that hash to location x continue to try the same sequence of alternate locations. If something has primary clustering, it will also have secondary clustering.

If a hash method does not exhibit primary or secondary clustering, it is termed non-clustering. This happens with double hashing. Even items which hash to the same location do not continue to try the same sequence of alternate locations because each is using a different step value.

Notice that non-clustering methods are the best, and methods that exhibit primary clustering are the worst.

(a) Using csilm.usu.edu (CS3/Hashing Algorithm), using linear probing (as the Collision Resolution technique), create a cluster due to primary clustering. Show how the competition exists even for values which originally did NOT want the same location. Attach a screen shot of the cluster you create.

(b) Using csilm.usu.edu (CS3/Hashing Algorithm), using quadratic probing as the Collision Resolution technique), create a cluster due to secondary clustering. Show how the competition exists even for values which originally did want the same location (even though they have different values). Attach a screen shot of the cluster you create. Explain the values which are in the cluster (as it won't be physically obvious).

(c) Using csilm.usu.edu (CS3/Hashing Algorithm), using double hashing (as the Collision Resolution technique), and examine clustering. For example, insert multiple copies of the same value to see how the series of probes progresses. Is there clustering? Insert multiple copies of another item which wants to be placed at the same location. Explain why they don't cluster. Attach a screen shot of the cluster you create. Explain the values you entered which refused to cluster (as it won't be physically obvious).

(d) Using csilm.usu.edu (CS3/Hashing Algorithm), find a Hash Function which seems to cause collisions. Explain why you think there are more collisions than with the other hash functions. You can trace through the actual code being used as a hash function by selecting Miscellaneous/Show Code.

6. Consider the following collision resolution schemes. Indicate whether they are non-clustering, secondary clustering, or primary clustering.
 - a. You use linear probing, but always increment by 3 (rather than 1) in the event of a collision.
 - b. You have 10 completely different hash functions. If the first method produces a collision, you try the second, and so on until you find a spot or until all hash functions are exhausted. At that point, the hash fails and you will need to rehash.
 - c. You have an overflow area at the bottom of the table. Instead of using open addressing (placing items that won't fit in the computed address in an open space), you put all items that won't fit in the computed address in this overflow area (sequentially).