1. Draw the 7-item hash table resulting from hashing the keys 2, 4, 7, 11, 12, 16, 20 using the hash function $h(i) = (2i + 1) \bmod 7$ and assuming collisions are handled with open addressing and linear probing.
   (**6 points**)
   Solution:
   The hash table will have the entries $\langle 20, 7, 4, 11, 12, 2, 16 \rangle$.

2. Draw a binary tree with height 4 and maximum number of external nodes. Is this tree unique?
   (**6 points**)
   Solution:
   Draw a balanced binary tree of height 4. It will have $2^4 = 16$ nodes and is unique.

3. Complete the pseudocode for the remove(p) operation in a doubly-linked list, where a node p is deleted and previous and next pointers are updated. What is the worst-case asymptotic running time?
(**5 points**)

Algorithm remove(p):

$t \leftarrow p$.element

(p.prev).next $\leftarrow p$.next /* linking out p */

(p.next).prev $\leftarrow p$.prev /* linking out p */

p.prev $\leftarrow$ null/* invalidating the position p */

p.next $\leftarrow$ null /* invalidating the position p */

return t

Running time: O(1)

4. Describe how to implement a stack using two queues. What is the running time of the push() and pop() methods in this case?
(**6 points**)
Solution:
Consider two FIFO queues Q_1 and Q_2. For a push operation into stack S, do an enqueue to Q_1. This takes $O(1)$ time. For a pop operation, dequeue all the elements in Q_1, return the last element to be dequeued, enqueue them to Q_2, and swap Q_1 and Q_2. In the worst case, a pop operation could take $O(n)$ time because $O(n)$ might need to be dequeued and enqueued.

5. Show that the problem of finding the k^{th} smallest element in a heap takes at least $\Omega(k)$ time in the worst case.
(**6 points**)
Solution:
To find the k^{th} smallest element, we will require k comparisons, because we do not know which path to take starting from the root node.

We can also perform $k-1$ extract-min operations on the heap to locate the k^{th} smallest element. In the best case, we can restore heap order in $O(1)$ time after each extract-min. Thus, finding the k^{th} smallest element takes $\Omega(k)$ time.

6. Suppose we perform a DeleteMin operation on the min heap $H = [1, 2, 3, 5, 6, 8, 11, 15]$ (the heap is stored here implicitly in the form of an array). Show the steps performed after deletion to restore the heap order of elements.
(**5 points**)
Solution:
Please see notes/slides.

7. Insert items with the following keys (in the given order) into an initially empty binary search tree: 30, 40, 50, 24. Draw the tree that results after each insertion.
   (**6 points**)
   Solution:
   Please see notes/slides.