Please respect the following guidelines for writing pseudocode:

1. C instructions are fine. But do not write object-oriented additions. Do not declare or use any class. Declare only procedures (if necessary) and explain in words what each procedure does, and what is the use of each parameter.

2. One instruction per line

3. Match the brackets with a horizontal line

4. Number your lines

5. Write down if your array is indexed 0 . . . n − 1 or 1 . . . n.

Please be aware that the last problem has four parts.

Problem 1 Assume the priority queue also must support the update DELETE(A, i), where i gives you the location in the data structure where element i is located. In a binary heap, that would be A[i]. Write pseudocode to achieve DELETE(A, i) in O(log n) time for binary heaps, where n is the size of the heap.

Problem 2 Using only the definition of a binary search tree, show that if a node in a binary search tree has two children, then its successor has no left child and then its predecessor has no right child.

Here all the keys in the binary search tree are distinct. The successor of a node j is defined as follows: if the node j has the biggest key, the successor is NIL; otherwise the successor is the node i that has a key that is bigger than the key of j and such that no other node has key between the key of i and the key of j.

Problem 3 Consider the following sequence of keys:

(5, 16, 22, 45, 2, 10, 18, 30, 50, 12, 1)

Consider the insertion of items with this set of keys, in the order given, into an initially empty (2,3) tree T. Draw T after each insertion.

Problem 4 Professor DontGoogleMyName claims that a (2,3) tree storing a set of items will always have the same structure assuming there are no deletions, regardless of the order in which the items are inserted. Show that Professor DontGoogleMyName is wrong, by giving two orders of the same set of items giving distinct trees. Draw the trees after each insertion.

Problem 5

1. Draw the 11-item hash table resulting from hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, and 5, using the hash function h(i) = (2i + 5) mod 13 and assuming collisions are handled by chaining.

2. What is the result of the previous exercise, assuming collisions are handled by linear probing?
3. Show the result of Exercise 1 above in this problem, assuming collisions are handled by quadratic probing, up to the point where the method fails because no empty slot is found. With the notation from the textbook, use $h$ itself as the auxiliary function $h'$ and $c_1 = 0, c_2 = 1$. Also, the key will not be inserted if no empty slot is discovered with at most 10 probes.

4. What is the result of Exercise 1 above in this problem, assuming collisions are handled by double hashing using a secondary hash function $h'(k) = 7 - (k \mod 7)$? With the notation from the textbook, use $h$ itself as the auxiliary function $h'$. 