1. Consider the following method to sort an array A of n integers, which we call BSTSort.

i) BST T;
ii) for (int i=0; i<n; i++)
iii) T.insert(A[i]);
iv) int i=0;
v) inorder(T.root);

where the method inorder is defined as

vi) public static void inorder(Node p) {
  vii) if (p=null) return;
viii) inorder(p.left);
ix) A[i++] = p.value;
x) inorder(p.right); }

(a) What is the worst case run-time of BSTSort?
(b) What is the average case behavior? Explain.
(c) Is it possible that instead of performing a one-by-one insertion in lines i-iii, we could replace it with something more efficient? For example, perhaps in the future someone will discover a way to build a BST in time O(n). (By way of analogy, we do not build a heap in such a way, as there is an O(n) build-heap method.)

[12 points]

2. Into an initially empty red-black tree

(a) insert the values 41, 38, 31, 12, 19, 8, 15, 5, 10, 6, 4, 2
(b) then delete the root

[13 points]

3. Write a (recursive?) routine which will take a RB tree and determine its black-height. What is its run-time? [8 points]

4. Consider an array containing the values 9, 5, 3, 13, 1, 2, 12, 8, 5, 16, 11, 4, 7, 6, 14, 10 in locations 1-16. Illustrate the build-heap method converting this array into a max-heap. [10 points]
5. Into an initially empty binomial heap (which, as in the text, is a MIN-heap):
   (a) insert the values 4, 7, 9, 2, 12, 8, 3, 10, 15, 21, 11
   (b) then remove the min.

[12 points]

6. Write a recursive routine which, given a BST T and integer k, will traverse T and print the contents of all nodes at depth k, from left to right. [10 points]

7. Given an array A on n integers and an integer k, we want to find the k-1 elements of rank \( \lceil n/k \rceil, 2 \lceil n/k \rceil, 3 \lceil n/k \rceil, \) and so on.
   (a) What is the run-time of the following?
      
      ```
      for (int i=1; i<k; i++)
          print A.select(i * ceiling(n/k))
      ```

   (b) Our goal is to do this in \( O(n \lg k) \) expected time. Rewrite Quicksort so that it stops when it locates an item of rank \( i \lceil n/k \rceil \).

   (c) Argue (no need to prove) why this might run in time \( O(n \lg k) \).

[15 points]

Total: 80 points