1. (5 points) Show the red-black tree that results after successively inserting the characters ‘Z’, ‘W’, ‘Q’, ‘G’, ‘N’ and ‘A’. For full credit, draw the tree after each insertion.

2. (5 points) Prove that if a black node in a red-black tree has just one child, that child must be red.

3. (5 points) Show the result of inserting 10, 12, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, and 2, one at a time, into an initially empty min binary heap and then removing the minimum element.

4. (10 points) In class we described the min binary heap implementation of the min priority queue ADT that provides for O( lg N ) performance for insert( x ), deleteMin() and O( 1 ) performance for findMin(). We also briefly discussed the max binary heap implementation of the max priority queue ADT that supports insert( x ), deleteMax() with O( lg N ) performance and findMax( ) with O( 1 ) performance.

Now consider the MEDIAN priority queue ADT that supports insert( x ) and deleteMedian( ) with O( lg N ) performance and findMedian( ) with O( 1 ) performance.

Describe how you would implement the median PQ.
   a. How would the data be stored?
   b. Describe the algorithm for insert( x )
   c. Describe the algorithm for deleteMedian( )
   d. Describe the algorithm for findMedian( )

Do NOT write the code for these methods.

Hint: consider using two binary heaps.