1. True or False. [5 points each]
   (a) F Computers are so fast today that complexity theory is only of theoretical, but not practical, interest.
   (b) T \( \log^* (2^n) = \Theta(\log^* n) \)

2. Fill in the blanks.
   (a) [10 points] Any comparison-based sorting algorithm on a file of size \( n \) must execute at least \( \Omega(n \log n) \) comparisons in the worst case. Alternative answer: \( \log_2(n!) \).
   (b) [10 points] Name two well-known divide-and-conquer sorting algorithms.
      quicksort
      mergesort

3. The answer to each of the following two questions is bubblesort, selection sort, or insertion sort.
   (a) [10 points] Treesort is a fast form of insertion sort.
   (b) [10 points] The following C++ code fragment implements selection sort. Assume that A is an array of size n, and that swap(int & a, int & b) transposes a and b.
      ```cpp
      for(int i = 0; i < n; i++)
      for(int j = i+1; j < n; j++)
      ```

4. Find the time complexity of each of these code fragments in terms of \( n \), using \( \Theta \) notation.
   (a) for(int i = 0; i*i < n; i++)
      \( \Theta(\sqrt{n}) \)
   (b) for(int i = 0; i < n; i++)
      for(int j = 1; j < i; j = 2*j);
      \( \Theta(n \log n) \)
   (c) for(int i = 1; i < n; i++)
      for(int j = i; j < n; j = 2*j);
      \( \Theta(n) \)
   (d) for(float x = n; x > 2.0; x = sqrt(x)) \quad (\text{sqrt}(x) \text{ returns the square root of } x.)
      \( \Theta(\log \log n) \)
(e) for(int i = 1; i < n; i = 2*i)
    for(int j = 2; j < i; j = j*j);

(Hint: use substitution)

Θ(log n log log n)

5. Solve the recurrences. Give asymptotic answers in terms of n. [10 points each.]

(a) \( F(n) = F(n/2) + 1 \)
    \( \Theta(\log n) \)

(b) \( F(n) = 2F(n/2) + 1 \)
    \( \Theta(n) \)

(c) \( F(n) = 2F(n/2) + n \)
    \( \Theta(n \log n) \)

(d) \( F(n) = n + F(n/5) + F(7n/10) \)
    \( \Theta(n) \)

(e) \( F(n) = F(3n/4) + F(n/2) + 3F(n/4) + n \)
    \( \Theta(n^2) \)

(f) \( F(n) = F(n-1) + \sqrt{n} \)
    \( \Theta(n^{3/2}) \)

(g) \( F(n) = F(n - \log n) + \log n \)

(h) \( F(n) = F(\log n) + 1 \)
    \( \Theta(\log^* n) \)

6. [20 points] A minheap is implemented as a binary tree, which is implemented as an array, as we saw in class. Suppose the array has 7 items as shown in the first row of the matrix below. Work out the steps of the minheap when the letter B is inserted. (Not all rows are needed.)

<table>
<thead>
<tr>
<th>A</th>
<th>C</th>
<th>J</th>
<th>D</th>
<th>F</th>
<th>R</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
<td>J</td>
<td>D</td>
<td>F</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>J</td>
<td>C</td>
<td>F</td>
<td>R</td>
<td>P</td>
</tr>
</tbody>
</table>

7. [20 points] Consider the following procedure:

void hold(int n)
{
    if(n >= 1)
    {
        for (int i = 1; i < n; i++)
            cout << "Thank you for holding. Someone will be with you shortly." << endl;
        hold(n-1);
    }
Consider the question of how many lines of output the execution of \texttt{hold(n)} would produce. Write down an appropriate recurrence for this question, and give an asymptotic solution in terms of \( n \), using either \( O \), \( \Omega \), or \( \Theta \), whichever is most appropriate.

\[ T(n) = 2T(n-1) + n \]
\[ T(n) = \Theta(2^n) \]

8. [30 points]

(a) Illustrate the circular linked list implementation of \texttt{queue}. Assume that the contents of the queue (from front to rear) is the list \texttt{Ann, Ted, Sue, Bob}.

(b) Illustrate the steps for inserting the item \texttt{Fay}.

(c) Illustrate the steps for deletion of an item, starting from the queue you illustrated for (8a).