1. Solve the following recurrence relations
   a. \( T(n) = T(n-1) + \frac{n}{2}, \ T(1) = 1 \)
   b. \( T(n) = 8T(n-1) - 15T(n-2), \ T(1) = 1, \ T(2) = 4 \)

2. Give asymptotic upper and lower bounds for \( T(n) \) in each of the following recurrences. Assume that \( T(n) \) is constant for \( n \leq 2 \). Make your bounds as tight as possible, and justify your answers.
   a. \( T(n) = 2T(n/2) + n^3 \)
   b. \( T(n) = T(9n/10) + n \)
   c. \( T(n) = 16T(n/4) + n^2 \)
   d. \( T(n) = 3T(n/2) + n \log n \)
   e. \( T(n) = T(n-1) + \log n \)

3. Argue that the solution to the recurrence \( T(n) = T(n/3) + T(2n/3) + n \) is \( \Omega(n \log n) \) by appealing to a recursion tree.

4. Use a recursion tree to solve the recurrence \( T(n) = T(\alpha n) + T((1-\alpha)n) + n \), where \( \alpha \) is a constant in the range \( 0 < \alpha < 1 \).

5. Prove that \( T(n) \), which is defined by the recurrence relation
   \[
   T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor \right) + 2n \log_2 n, \ T(2) = 4, 
   \]
   Satisfies \( T(n) = O(n \log^2 n) \)

6. Compare the following pairs of functions in terms of order of magnitude. In each case, say whether \( f(n) = O(g(n)) \), \( f(n) = \Omega(g(n)) \), and/or \( f(n) = \Theta(g(n)) \)

<table>
<thead>
<tr>
<th>f(n)</th>
<th>g(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. 100n + \log n</td>
<td>n + (\log n)^2</td>
</tr>
<tr>
<td>b. \log n</td>
<td>\log(n^2)</td>
</tr>
<tr>
<td>c. \frac{n^2}{\log n}</td>
<td>n(\log n)^2</td>
</tr>
<tr>
<td>d. \log n^{\log n}</td>
<td>n/\log n</td>
</tr>
<tr>
<td>e. \sqrt{n}</td>
<td>(\log n)^5</td>
</tr>
<tr>
<td>f. n 2^n</td>
<td>3^0</td>
</tr>
</tbody>
</table>
7. Answer questions 1a to 1c pertaining to the STOOGE_SORT Algorithm given below.

STOOGE_SORT(A,i,j)

1) if $A[i] > A[j]$
2) then exchange $A[i] \leftrightarrow A[j]$
3) if $i+1 \geq j$
4) then return
5) $k \leftarrow \lfloor (j-i+1)/3 \rfloor$
6) STOOGE_SORT(A,i,j-k)
7) STOOGE_SORT(A,i+k,j)
8) STOOGE_SORT(A,i,j-k)

a. Argue that STOOGE_SORT (A,1,length[A]) correctly sorts the input array 
   $A[1..n]$, where $n = \text{length}[A]$.

b. Give a recurrence for the worst-case running time of STOOGE_SORT and a tight 
   asymptotic bound on the worst-case running time.

c. Compare the worst-case running time of STOOGE_SORT with that of insertion
   sort, merge sort, heapsort, and quicksort.

8. Given an array of integers $A[1..n]$, such that, for all $i$, $1 \leq i < n$, we have
   algorithm to find $j$ such that $A[j] = z$ for a given value $z$, $x \leq z \leq y$. What is the maximal
   number of comparisons to $z$ that your algorithm makes?

9. What is Counting Sort? Give the Algorithm and provide analysis.

10. What are Fibonacci Heaps? Give an example of a Fibonacci Heap. Write
    basic algorithms for insertion and deletion in Fibonacci Heaps.

11. You are given a collection of $n$ bolts of different widths and $n$ corresponding
    nuts. You are allowed to try a nut and bolt together, from which you can
    determine whether the nut is larger than the bolt, smaller than the bolt, or
    matches the bolt exactly. However there is no way to compare two nuts
    together or two bolts together. The problem is to match each bolt to its nut.
    Design an algorithm for this problem with average case efficiency of $\Theta(n \log n)$. 
12. Explain how we can check a graph’s a cyclicity by using Breadth-first search. Does either of the two traversals – DFS or BFS – always find a cycle faster than the other? If your answer is yes, indicate which of them is better and explain why it is the case; if you answer no, give two examples supporting your answer.

13. A graph is said to be bipartite if all its vertices can be partitioned into two disjoint subsets X and Y such that every edge connects a vertex in X with a vertex in Y.
   a. Design a DFS-based algorithm for checking whether a graph is bipartite
   b. Design a BFS-based algorithm for checking whether a graph is bipartite.

14. Prove that the number of different paths of length $k > 0$ from the $i$th vertex to the $j$th vertex of a graph (undirected or directed) equals the $(i,j)$th element of $A^k$, where $A$ is the adjacency matrix of the graph. [to be discussed in class].

15. You are given a list of numbers for which you need to construct a min-heap. (A Min-heap is a complete binary tree in which every key is less than or equal to the keys in its children.) Write a min-heap algorithm and analyze its complexity.

16. Suppose we are to find the $k$ smallest elements in a list of $n$ elements, and we are not interested in their relative order. Can a linear-time algorithm be found when $k$ is a constant? If so give the algorithm. In either case, justify your answer.

17. Modify Dijkstra’s single source shortest path algorithm so that it checks if a directed graph has a cycle. Analyze your algorithm.