1. For the functions satisfying the following three recurrences, determine which is the fastest growing and which is the slowest growing
   (a) $T(n) = 2T(n-1), T(1) = 5.$
   (b) $S(n) = S(n-1) + \lceil \log(n) \rceil, S(1) = 17.$
   (c) $R(n) = 3R(\lfloor n/2 \rfloor) + 4, R(1) = 11.$

2. Prove that $\log^2(n) \in O(2^{\sqrt{n}}).$

3. Give a proof by invariants that the following algorithm correctly searches a sorted list $X[0], \ldots, X[n-1]$ for element $z$.

   ```
   Search(X,n,z) {
       i = 0
       j = n-1
       while (i < j) {
           k = (i+j)/2
           if (X[k] = z)
               return(k)
           if (X[k] < z)
               i = k+1
           else j = k-1
       }
       if (X[i] = z)
           return(i)
       else return(Failure)
   }
   ```

   Assumption: $X[1] \leq X[2] \leq \cdots \leq X[n-1]$

4. Let $H$ be an array containing a min-heap with $n$ elements (the smallest element is at the top, and the children of $H[i]$ are $H[2i]$ and $H[2i+1]$). Give pseudocode for an efficient algorithm for deleting the smallest element from $H$. What is the worst case time complexity of your algorithm?
5. (a) Find a sharp upper bound on the height of a 2-3 tree with $n$ nodes.
(b) Compare the performance of search trees, red-black trees, B-trees, or other such structures under various mixes of the basic operations (search, insert, delete).

6. Describe an efficient algorithm to count the connected components in an undirected graph. Analyze the complexity of your algorithm. It should run in linear time.

7. Let $G = (V, E)$ be a directed graph with edge capacities $c(u, v) \geq 0$, a sink $s$, and a source $t$. Prove that if $f$ is a flow on $G$ and $f'$ is a flow on the residual graph $G_f$, then the function $g$ defined by $g(u, v) = f(u, v) + f'(u, v)$ is a flow on $G$.

8. Consider the following problem:

**Instance:** An integer $K \geq 0$ and $n$ distinct positive integers $x_1, \cdots, x_n$.

**Output:** A subset of $\{x_1, \cdots, x_n\}$ whose sum is precisely $K$ (if there is such a subset).

Design an $O(nK)$ algorithm that solves this problem. Give pseudocode, explain your pseudocode, and analyze the algorithm. (HINT: Use dynamic programming).

9. (a) Give pseudocode for and explain the Euclidean algorithm (which finds the GCD of two integers).

(b) Analyze the time complexity of the Euclidean algorithm.

(c) Show that the problem of testing whether an integer is not a prime number is in the complexity class NP.

10. (a) Define the term “polynomial time mapping reduction.”
(b) HITTING SET is the problem:

**Instance:** A collection $C = \{C_1, \cdots, C_m\}$ of subsets of a set $S$ (i.e., each $C_i \subseteq S$), and a positive integer $k$.

**Question:** Does $S$ contain a subset $T$ such that $|T| \leq k$ and $T$ contains at least one element of each $C_i$?

Prove that HITTING SET is NP-complete. (Hint: use the fact that VERTEX COVER is NP-complete).