CSE 317: Design and Analysis of Algorithms
Sample Questions: Midterm

1. *O*-notation questions. Prove/disprove:

    (a) If $f(n) = O(g(n))$ and $g(n) = O(h(n))$ then $f(n) = O(h(n))$.

    (b) If $f(n) = O(g(n))$ then $f(n) = o(g(n))$.

    (c) If $f(n) = o(g(n))$ and $f(n) = \omega(h(n))$ then $h(n) = o(g(n))$.

2. Solve recurrences:

    (a) $A(n) = 3T\left(\dfrac{n}{4}\right) + n \log n$ with $A(1) = 1$.

    (b) $B(n) = 3B(n-1) - 2B(n-2)$ with $B(0) = 1$ and $B(1) = 2$.

    (c) ** $C(n) = 2C(\sqrt{n}) + 3\log_2 n$ when $n > 2$ and $C(n) = 1$ for $n = 2$.

3. Let G be a connected undirected graph with distinct edge weights. Prove that the MST is unique.

4. In any connected graph, removing the heaviest edge from the MST always increases its weight.

5. The minimum edge incident to every vertex must appear in every MST.

6. Suppose a graph has negative edge weights but no negative cycles. Does Prim's algorithm still work correctly? Explain.

7. Given two sorted arrays A and B, each of length n, find the median of all elements of $A \cup B$ in $O(\log n)$ time.

8. Given sorted arrays A and B, find whether there exists $a \in A, b \in B$ such that $a - b = k$. Achieve $O(n)$ time.

9. Given two sorted arrays A, B, find the k-th smallest sum among all $a + b$ pairs.

10. Given a sorted array A, count the number of triplets (i, j, k) such that $A[i] + A[j] = A[k]$.

11. Given an $n \times n$ matrix where each row and column is sorted, design an $O(n)$-time search algorithm for an element x.

12. Modify your algorithm if the matrix rows are sorted ascending but columns descending.

13. Devise a divide-and-conquer algorithm to compute the transpose of a large square matrix in parallel. Analyze work and depth.

14. Write the recurrence for the Longest Common Subsequence (LCS) and analyze time complexity.