Midterm Examination

Time allowed: 90 minutes. Total: 150 points.

1. (10 points) Give the worst-case time complexity of searching for a key in a balanced binary search tree with n nodes, and justify your answer in one sentence.

   Solution: $O(\log n)$: a balanced tree has height $O(\log n)$, and a search follows one root-to-leaf path.

2. (20 points) Consider the following recurrence for a divide-and-conquer algorithm:

   $$T(n) = 3T(n/2) + O(n).$$

   (a) (5 points) Identify a, b, and $f(n)$ for the Master Theorem.

   Solution: $a = 3$, $b = 2$, $f(n) = O(n)$.

   (b) (10 points) Compute $n^{\log_b a}$ and compare it to $f(n)$ to determine which case of the Master Theorem applies.

   Solution: $n^{\log_2 3} \approx n^{1.585}$, which grows strictly faster than $f(n) = O(n)$, so this is Case 1.

   (c) (5 points) State the resulting closed-form bound on $T(n)$.

   Solution: $T(n) = \Theta(n^{\log_2 3})$.

3. (30 points) A hash table with $m = 16$ slots uses separate chaining and currently holds $n = 12$ keys.

   (a) (10 points) What is the load factor α of this table?

   Solution: $\alpha = n/m = 12/16 = 0.75$.

   (b) (10 points) Under simple uniform hashing, what is the expected length of a chain in this table?

   Solution: The expected chain length equals the load factor, $\alpha = 0.75$.

   (c) (10 points) The table is resized to $m = 32$ slots once α exceeds 0.75. After the next insertion, will a resize be triggered? Explain.

Solution: Yes: after the next insertion $n = 13$, giving $\alpha = 13/16 = 0.8125 > 0.75$, so the resize threshold is crossed.

4. (15 points) Short answer: why does inserting into a min-heap run in $O(\log n)$ time rather than $O(n)$?

Solution: A heap is a complete binary tree of height $O(\log n)$; inserting appends a new leaf and "sifts up" along at most one root-to-leaf path, so the number of swaps is bounded by the tree height.

5. (25 points) Given the adjacency list of a graph with V vertices and E edges, state the running time of breadth-first search and explain the role of each term.

Solution: $O(V + E)$: every vertex is enqueued and dequeued at most once ($O(V)$), and every edge is examined at most once (twice for an undirected graph, still $O(E)$) when scanning each vertex's adjacency list.