Z5007: End-Semester Comprehensive Mock Exam & Solutions

Programming and Data Structures (Weeks 1–17) · Odd Semester 2026-27 · innocent@iitmz.ac.in

Exam Format: Time: 3 Hours · Closed Book · Total: 100 Marks. Attempt all three sections before inspecting the worked solutions on Page 2.

Part A: Asymptotic Analysis, Heaps & Hash Tables [25 Marks]

A1. **[6 marks]** Solve $T(n) = 3T(n/2) + n \log n$ using the Master Theorem, stating a, b, $\log_b a$, and the tight Θ-bound.

A2. **[6 marks]** Starting with an empty max-heap, insert keys $12, 5, 19, 8, 25, 14$ in order. Write the resulting 0-indexed array and trace one `extract_max()` call.

A3. **[6 marks]** Keys $\{22, 35, 14, 28, 42\}$ are inserted into a hash table of size $m = 7$ using $h(k) = k \bmod 7$ and linear probing. Show the final table, then explain what goes wrong if key 14 is deleted by setting its slot to `EMPTY`.

A4. **[7 marks]** Insert $30, 10, 20$ into an empty AVL tree. Identify the imbalance type at 30 after inserting 20 and draw the balanced tree after rotations.

Part B: Graphs, Shortest Paths & Minimum Spanning Trees [35 Marks]

B1. **[10 marks]** A weighted directed graph has vertices $\{S, A, B, C, T\}$ and edges $S \to A(2)$, $S \to B(5)$, $A \to B(1)$, $A \to C(6)$, $B \to C(2)$, $C \to T(3)$. Trace Dijkstra's algorithm from S, recording distance and predecessor updates at each pop.

B2. **[10 marks]** Give a minimal 3-vertex example showing why Dijkstra fails on negative edge weights, and state the exact worst-case runtime of Bellman-Ford on $|V|$ vertices and $|E|$ edges.

B3. **[15 marks]** Explain how Kruskal's algorithm uses Disjoint Set Union (path compression + union by rank) to maintain acyclicity, and prove using the Cut Property why the lightest edge crossing $(S, V \setminus S)$ belongs to an MST.

Part C: Dynamic Programming, Strings & NP-Completeness [40 Marks]

C1. **[14 marks]** Given 0/1 Knapsack items $(w, v) = \{(2, 6), (3, 10), (4, 12)\}$ and capacity $W = 5$, fill the 1D DP array backwards for each item and recover both optimal value and chosen items.

C2. **[12 marks]** Compute the KMP prefix table $\pi[0..6]$ for pattern $P =$ `"ABABACA"` and explain why KMP search runs in $O(n + m)$ worst-case time.

C3. **[14 marks]** Define **NP-hard** vs. **NP-complete**, and give a polynomial-time reduction from `Independent Set` to `Vertex Cover`.

Worked Marking Key & Step-by-Step Solutions

Part A Solutions:

A1. $a = 3, b = 2$, so $\log_2 3 \approx 1.585$. Since $f(n) = n \log n = O(n^{1.585-\epsilon})$ for $\epsilon = 0.5$, Case 1 applies: $T(n) = \mathbf{\Theta(n^{\log_2 3})}$.

A2. After inserts: $12 \to [12]$; $5 \to [12, 5]$; $19 \to [19, 5, 12]$; $8 \to [19, 8, 12, 5]$; $25 \to [25, 19, 12, 5, 8]$; $14 \to [\mathbf{25}, \mathbf{19}, \mathbf{14}, \mathbf{5}, \mathbf{8}, \mathbf{12}]$. On `extract_max()`: return 25, move 12 to root $[12, 19, 14, 5, 8]$, sift down swapping $12 \leftrightarrow 19$, then 12 stays above $5, 8$: $[\mathbf{19}, \mathbf{12}, \mathbf{14}, \mathbf{5}, \mathbf{8}]$.

A3. Hash slots mod 7: $22 \to 1$, $35 \to 0$, $14 \to 0$ (probes to 2), $28 \to 0$ (probes to 3), $42 \to 0$ (probes to 4). Table: $[35, 22, 14, 28, 42, _, _]$. Setting slot 2 to `EMPTY` breaks the probe chain for 28 and 42, causing lookups to fail with false misses; place a `TOMBSTONE` marker instead.

A4. Inserting 30, left child 10, right child of $10 = 20$ creates an **LR imbalance** at 30 ($bf(30) = +2, bf(10) = -1$). Left rotate at 10 (giving $30 \leftarrow 20 \leftarrow 10$), then right rotate at 30: root **20**, left child **10**, right child **30**.

Part B Solutions:

B1. Init: $d[S] = 0$, others ∞. Pop $S(0)$: $d[A] = 2(S), d[B] = 5(S)$. Pop $A(2)$: relax $A \to B$ gives $d[B] = \min(5, 2+1) = 3(A)$, $d[C] = 8(A)$. Pop $B(3)$: relax $B \to C$ gives $d[C] = \min(8, 3+2) = 5(B)$. Pop $C(5)$: $d[T] = 8(C)$. Pop $T(8)$. Shortest path: $S \to A \to B \to C \to T$ (total cost **8**).

B2. Graph $S \to A(1)$, $S \to B(2)$, $B \to A(-3)$: Dijkstra pops A first at distance 1 and locks it permanently, missing $S \to B \to A = -1$. Bellman-Ford runs $V - 1$ passes over all E edges in $\mathbf{O(VE)}$ time.

B3. Kruskal sorts edges ascending by weight and checks `find(u) != find(v)` in $O(\alpha(V))$ amortized time before calling `union(u,v)`. Cut Property: if MST T omits lightest crossing edge $e = (u, v)$, adding e creates a cycle with another crossing edge e' ($w(e') \geq w(e)$); swapping e' for e yields an MST containing e.

Part C Solutions:

C1. Init $dp = [0, 0, 0, 0, 0, 0]$. Item $(2, 6)$ ($w = 5 \ldots 2$): $[0, 0, 6, 6, 6, 6]$. Item $(3, 10)$ ($w = 5 \ldots 3$): $dp[5] = \max(6, 10+6) = 16, dp[4] = 10, dp[3] = 10 \to [0, 0, 6, 10, 10, 16]$. Item $(4, 12)$ ($w = 5 \ldots 4$): $dp[5] = \max(16, 12 + 0) = 16, dp[4] = 12 \to [0, 0, 6, 10, 12, 16]$. Optimal value = **16** from items $\{(2, 6), (3, 10)\}$.

C2. Pattern `ABABACA`: $\pi = [\mathbf{0}, \mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{0}, \mathbf{1}]$. Because the text cursor advances monotonically and total prefix-table fallbacks are bounded by text advancements n, construction takes $O(m)$ and search takes $O(n)$, totaling $\mathbf{O(n + m)}$.

C3. A problem L is **NP-hard** if every $A \in \mathbf{NP}$ reduces $A \leq_p L$; L is **NP-complete** if L is both NP-hard and in **NP**. Reduction IS $\leq_p$ VC: given $(G = (V, E), k)$, output $(G, |V| - k)$ unchanged. A subset $I \subseteq V$ has no internal edges iff every edge in E has at least one endpoint in $V \setminus I$; thus G has an independent set of size k iff G has a vertex cover of size $|V| - k$.