CS5800 – Dynamic Programming Problems: Comparison

Generated 2026-06-18. Legend: Θ tight bound, O upper, Ω lower; $\to$ arrow; $\gg$ much greater; $\star$ starred problem. Shaded rows = interval-DP family.

Problem	Complete /10	RT & Mem (DP, bottom-up)	Better bottom-up RT / Mem	Memo skips subproblems?	Memo helps with computation order?	Major bugs	Minor bugs	Related ideas
Coin change	10	$\Theta(nk)$ time, $\Theta(n)$ space; Make-Change $O(n)$	$\Theta(nk)$ tight; space $\to \Theta(d_k)$ if value-only	No — d_1=1 $\Rightarrow$ all 0..n reachable	No — trivial (increasing amount p)	None	Recursive digression hardcodes $\{1, 5, 10\}$	Unbounded knapsack (min-cost); DAG shortest path ($p \to p-d_i$)
Checkerboard	9	$\Theta(mn)$ time, $\Theta(mn)$ space; Check-Sol $\Theta(m)$	$\Theta(mn)$ tight; space $\to \Theta(n)$ (one row) if value-only	No — answer = min over whole last row	No — easy (row 1 $\to m$)	Borderline: recurrence wrong at j=1, j=n (off-board reads); needs ∞ guard	i-loop should start at 2 (reads/clobbers row 0); min vs argmin (Check-Sol L1); prints row 0 (downto 2); $(l, k) \to (k, l)$	Layered-DAG shortest path; grid/triangle min-path-sum; Viterbi
Knapsack (0/1)	10	$\Theta(nW)$ time; space *stated* $\Theta(n^2)$ [**wrong**: $\Theta(nW)$]; Items $O(n)$	$\Theta(nW)$ tight; space $\to \Theta(W)$ rolling row; Hirschberg traceback	**Yes** — only w=$W-$(subset sum) visited; skips much of $\Theta(nW)$ when $W \gg$ #weights	No — easy (item by item, row-major)	**Incorrect claim:** space $\Theta(n^2)$ (should be $\Theta(nW)$)	ITEMS print / $W-=w[n]$ not nested under `if`; "$0 \le i, w \le n$" should be $w \le W$	Subset-sum / partition (v=w); unbounded = coin change; fractional = greedy
LCS	10	$\Theta(mn)$ time, $\Theta(mn)$ space; Print-LCS $O(m+n)$	length-only space $\Theta(\min(m, n))$; Hirschberg; time $O(mn/\log n)$ Four-Russians; bit-parallel $O(mn/w)$	Barely — worst case (few matches) reaches all mn	No — easy (row-major)	None	"proceeds" $\to$ "precedes"	Edit / Levenshtein distance; Needleman–Wunsch; LIS = LCS(A, sorted A)
Max-duration scheduling (weighted interval)	10	$O(n \log n)$ (linear DP + 2 sorts); $O(n)$ if keys sort linearly; space $O(n)$	$O(n \log n)$ tight in comparison model; $O(n)$ with radix / pre-sorted	No — skip-branch $dp[k+1]$ chains through all indices	Low — sort by start, then a simple right-to-left pass	None	Setup table lists G duration 3; 14$-$12=2 (Gantt + DP trace use 2)	Weighted interval scheduling; unit weights $\to$ activity selection (greedy); max-wt independent set on interval graph
Jars on ladder (egg drop / HSR)	N/A (∼9, unsolved spec)	MinT(n, k): $O(n^2k)$ time, $O(nk)$ space	dual MaxR$(k, q) \to O(nk)$, or $O(k \cdot$MinT$)$ via $\sum C(q, i)$; Knuth monotonicity $\to O(nk)$; space $\to O(k)$	No — inner min over all first rungs requests every smaller $(j, k-1), (j, k)$	Low — nested increasing (k, n); MaxR by increasing q	None (statement)	MaxR(k, q) in B vs MaxR(q, k) in C-hint (arg-order flip)	Classic egg-drop; MinT $\leftrightarrow$ MaxR DP-dual; $\sum C(q, i) \ge n$ = decision-tree bound; ties to Optimal BST
Longest hike $\star$	N/A (∼9, unsolved spec)	$\Theta(n^2)$ time, $\Theta(n^2)$ space; row-major fill **invalid**	Part A (ints $\le$ 29000): counting-sort buckets $\to \Theta(n^2)$. Part B (reals): Kahn/queue topological (BFS) $\to \Theta(n^2)$, vs $\Theta(n^2 \log n)$ sort	No — max over all starts $\Rightarrow$ all n^2 cells needed	**Yes, strongly** — order is by altitude (not position); reals need BFS-queue or DFS-memo. The point of Part B	None (statement)	"he matrix" $\to$ "the"; "dont" $\to$ "don't"	Longest path in a DAG (Lec-8 DP-as-DAG); topological-order DP; +1 edges $\Rightarrow$ acyclic; toroidal grid; counting-sort to order states
Painters / fence (Split-Array-Largest-Sum)	∼8 (handwritten)	$\Theta(n^2k)$ time (prefix sums $\Rightarrow O(1)$/split); $O(nk)$ space $\to O(n)$ rolling	binary search on the answer $+O(n)$ feasibility $\to O(n \log(\sum L))$ (no k); or monotone-split / D&C $\to \sim O(nk \log n)$	No — $dp[i][p]$ all needed	Low — nested increasing (i, p)	None evident (handwritten)	—	LeetCode 410 Split-Array-Largest-Sum; book/file allocation; makespan w/ contiguous blocks; **minimax** (bottleneck) objective $\Rightarrow$ max inside min; binary-search-on-answer
Bitonic tour (CLRS 15-3 / Chris)	∼8 (handwritten)	CLRS 2-D $B[i, j]$: $\Theta(n^2)$ time, $\Theta(n^2)$ space; $+O(n \log n)$ sort	Chris's 1-D $C[m]$: $\Theta(n^2)$ time, $O(n)$ space (split j + consecutive-chain prefix sums); sort $O(n \log n)$	No — min over all splits requests the chain	Low — increasing m (Chris) / increasing j (CLRS)	None evident (handwritten)	(photo) $\sum$ upper limit reads "$i-1$", is "$R-1$"	Euclidean TSP (poly x-monotone special case); two-paths DP; CLRS 15-3 (14-3 4th ed); min-sum
Optimal BST (your notes / CLRS)	∼7 (handwritten sketch; HW6 Q1 finishes steps 3–5)	$\Theta(n^3)$ time, $\Theta(n^2)$ space (e, root, w tables)	Knuth's optimization $\to O(n^2)$ (root monotone; QI holds: $w(i, j)$ split-independent); space $\Theta(n^2)$	No — inner min over all roots requests every interval (like matrix chain)	Mild — by increasing interval length (diagonal), like matrix chain	None evident (handwritten)	(not symbol-audited)	Matrix chain (sibling; only OBST gets Knuth); polygon triangulation; Hu–Tucker / alphabetic codes; vs Huffman (greedy); $+w(i, j)$ = "every node sinks one level"
Matrix chain	10	$\Theta(n^3)$ time, $\Theta(n^2)$ space; Print $O(n)$	$O(n \log n)$ (Hu–Shing, polygon partitioning) — *not* Knuth (added weight depends on split k); space $\Theta(n^2)$	No — inner min over all splits requests every interval	Mild — by chain length (diagonal), not row-major	None	"k=$i, \ldots, j$+1" should be $j-1$ (claim itself correct); "splited / os / in and" typos	Optimal BST (sibling); = polygon triangulation (Hu–Shing); CYK

Interval-DP family (shaded): matrix chain, Optimal BST, bitonic — all split-at-k over an interval, fill diagonally, and let memoization skip nothing; the speedups differ — OBST $\to$ Knuth $O(n^2)$, matrix chain $\to$ Hu–Shing $O(n \log n)$, bitonic already $O(n^2)$ (Chris $\to O(n)$ space).

Painters is the lone minimax (bottleneck) DP — its recurrence nests *max* inside *min* rather than summing, which is exactly why binary-search-on-the-answer applies to it and not to the additive DPs.