15-451/651 Algorithm Design & Analysis, Spring 2026

Quiz 1 Solutions

Problem: There are n items with positive integer weights $w_1, \dots, w_n$ (one item of each weight). You want to make total weight W (also a positive integer) while minimizing the weight of the heaviest item used. Design an algorithm that given $w_1, \dots, w_n$ and W returns the weight of the heaviest item used, or reports that it is impossible to make total weight W using the items given. The runtime should be $O(nW)$.

Part 1: Definition of DP state (3 points). For $0 \le i \le n$ and $0 \le w \le W$, we define $DP[i][w]$ to be the minimum possible weight of the heaviest item, if we only use the first i items and the total weight is w.

Part 2: DP Transition with brief justification (4 points). If $w < w_i$ then $DP[i][w] = DP[i-1][w]$; you can't use the weight w_i. Otherwise,

$$DP[i][w] = \min\{DP[i-1][w], \max\{DP[i-1][w-w_i], w_i\}\}.$$

This is because we can either not use w_i (hence going to the state $DP[i-1][w]$, or you can use w_i, and the heaviest item is now the maximum of the heaviest item used in $DP[i-1][w-w_i]$ and w_i itself.

Part 3: Base cases (2 points). The base case is $DP[0][0] = 0$ and $DP[0][j] = +\infty$ for $j > 0$.

Part 4: Final answer (1 point). The answer is $DP[n][W]$. If it equals $+\infty$, then that means it was not possible to create total weight W.