Gabe McKay
Terisa Le
Reeve Masilamani

ECS 122A

Homework 2

October 23, 2025

Problem 4

$dp[i, j, k] \doteq (p, w)$ where w is extra weight of some CDs, cassettes, and vinyls that are in an open box with some space left over from items [1..i], [1..j], etc. p is the minimum number of packages needed to store 1..i, 1..j, and 1..k.

Recurrence:
$p, w \leftarrow \infty$
for t = 1 to 3:
  $w_{new}, p_{new} = dp[i-$(1 if t=1 else 0)$][j-$(1 if t=2 else 0)$][k-$(1 if t=3 else 0)]
  if $w_{new} + w_t \leq 1$:
    p, w = min((p, w), (p_{new}, $w_{new} + w_t$))
  else:
    p,w = min((p, w), ($p_{new} + 1$, w_t))

Note: The min to p,w minimizes p first, and if there is a tie, it then chooses the lowest w.

Algorithm:

for i = 1 to n_1:
  for j = 1 to n_2:
    for k = 1 to n_3:
      Recurrence
return dp[n_1][n_2][n_3].p + 1 (if $w > 0$)

Runtime: Three loops $n_1 * n_2 * n_3 * O(1)$ work from each call to the recurrence meaning $O(n_1 n_2 n_3)$ work overall.

When we get to some dp[i][j][k] we have filled all previous entries through our loop structure.
In our recurrence, we look at three cases. One where we take a CD and put it in a box, one where we take a cassette, ect.
We look for the minimum state between these actions to choose what the minimum state is dp[i][j][k].
This means that every dp[i][j][k] represents the lowest state possible so dp[n_1][n_2][n_3], represents the lowest state overall which is our final answer.