EECS 336 Problem 6.1

1. Identify subproblem

OPT(i, j) = minimum unhappiness you will get from day i to D when the remaining chapters are j to n.

2. Recurrence:

OPT(i, j) = $min_{T_i=\sum_{i=j}^{k} t_i, k\in\{j,\ldots,n\}}$ {OPT(i+1, k) + $[max(m_i - T_i, 0)]^4 + max(T_i - m_i, 0)$, OPT(i+1, j) + m_i^4}

3. Base case:

All the remaining chapters should be read in the last day:
OPT(D, j) = $max(m(D) - T, 0)^4 + max(T - m(D), 0)$, where $T = \sum_{i=j}^{n} t_i$ if there is any chapter remaining on the last day, and T=0 if no chapter is left.

4. Iterative Dynamic Programming

Algorithm 1 Reading and/or Weeping

Require: array $m = \{m_i\}$ (the reading minutes on each day #i) and $t = \{t_j\}$ (the time required to read chapter #j)

function RW(m, t)
  Set D = m.length, n = t.length
  Initialize Memo = new array[D][n+1]
  Set base case:
  $Memo[D][n+1] = (m[D])^4$
  for `j = 1 to n` **do**
    Memo[D][j] = $max(m(D) - T, 0)^4 + max(T - m(D), 0)$, where $T = \sum_{i=j}^{n} t_i$
  end for
  for `i = D-1 down to 1` **do**
    for `j = 1 to n+1` **do**
      Memo[i][j] = $min_{T_i=\sum_{i=j}^{k} t_m, k\in\{j,\ldots,n\}}$ {Memo(i+1, k) + $[max(m_i - T_i)]^4 + max(T_i - m_i, 0)$, Memo(i+1, j) + m_i^4}
    end for
  end for
  Return Memo[1][1]
end function

RunTime

1. Initialization takes O(n) time
2. Memo[ ][ ] is a two dimensional (D-by-(n+1) table), and each updates takes O(n) time, so the total runtime on updates is $O(Dn^2)$

Correctness of the recurrence

(Proof by induction)

Assume OPT(i, j), OPT(i, j+1) ... OPT(i, n+1) is the minimum total unhappiness you would get from day i to D with the remaining chapter {j to n},{j+1 to n}...{no remaining chapter}. Then OPT(i-1, k) (k $\leq$ j) should be the minimum among the set I, where I = all possible unhappiness gained on day i by reading chapter k to j, k to j+1...k to the end + the minimum unhappiness gained after day i with the remaining chapters. So the recurrence is correct.