EECS 336 Fall 2015
Homework Problem 6.1

i) Opt(i,j) = Minimum unhappiness of reading sections 1...j over days 0...i

ii) Opt(i,j) = $\min_{0 \le k \le j}(Opt(i-1,k) + S_i\{k+1...j\} + F_i\{k+1...j\}^4)$

F_i\{k+1...j\} is defined as the free time on day i after reading sections (k+1) through j
S_i\{k+1...j\} is defined as the lost sleep on day i after reading sections (k+1) through j

iii) Base Cases:
No unhappiness from reading sections 1...j within 0...0 days:
Opt(0, j) = 0

Infinite unhappiness from not finishing n sections of K&T by day D:
Opt(0,n) = ∞, Opt(D, j!=n) = ∞

iv) Here is the algorithm:

Algorithm 1 Minimum_Unhappiness

memo[0..D][0..n]
memo[0][j] = 0
memo[0][n] = ∞
for all $i \in$ 1...D **do** {iterate over all rows except base case}
  for all $j \in 0...n$ **do** {iterate over all columns}
    memo[i][j]= $\min_{0 \le k \le j}(memo[i-1,k] + S_i\{k+1...j\} + F_i\{k+1...j\}^4)$
return memo[D][n]

Correctness
To find the minimum unhappiness of reading up to sections 1..j over days 0...i, we minimize over two parameters: 1) the minimum unhappiness up to day i-1 for sections 0...k, with 2) free time or lost steep on day i reading sections k+1 to j. Obviously, from the given formulas, either F_i or S_i will be 0. Checking those to items for the prior day's reading, yields a correct recurrence and the basis for a working and efficient algorithm.

Runtime
This algorithm works by iterating over the memo table of size n x m, where each item does size n work (checks some or all of the row above). The run-time is therefore $\mathcal{O}(n^2 * m)$).This is much faster than a brute force method.

Code
Please see attached code & code printouts