I collaborated with:

Problem

Answer Exercise 6 (page 317) in Chapter 6 of your text. Include the requisite justification of correctness and complexity analysis. Also address the following: Why do you think the problem has you minimize the sum of the *squares* of the slack of each line, as opposed to the the sum of the slacks themselves or of the absolute values of the slacks.
By the way, in the sentence "The difference between the left-hand side and the right-hand side will be called the *slack* of the line..." the authors are referring to the left-hand side and the right-hand side of the inequality.
Hint: Wouldn't it be useful to know all possible potential slack values...?

Solution

Let $S_{i,j}$ denote the slack of a line starting at word w_i and ending at word w_j. We must start from the last line and figure out how many words appear on this line. We start from word w_n and check to see up to what word w_k we can fill this line where $k \leq n$. Note that $S_{k,n}$ takes $O(n)$ and we must find the minimum slack for said line. Once we find it we move on to find $S_{i,k-1}$ where $i \leq k-1$. We can build the following recurrence: $opt[n] = \min_{1 \leq k \leq n}(S^2_{k,n} + opt[k-1])$.

Algorithm 1 Determines partition of words

$opt[0] = 0$
for i from 1 to n **do**
  $opt[i] = \min_{1 \leq k \leq i}(S^2_{k,i} + opt[k-1])$
end for

We then simply trace back opt and find the optimal solution for an additional $O(n)$. The problem asks us to minimize the sum of the squares of slack because it emphasizes on more spaces for words on a line.
Time complexity: $O(n^2)$
Space Complexity: $O(n)$

Proof. Base Case: If the text consist of a sequence of 1 word, then that word should constitute the last line. Our algorithm will find the slack of this word and add $opt[0] = 0$ to it. This is the minimum possible slack for this sequence, so our algorithm works for this case.
Inductive Hypothesis: Assume our algorithm solves the problem for a sequence up to k words where $k \geq 1$.
Inductive Step: We must now show that our algorithm provides the best solution for a sequence of size $k+1$. Let W be a sequence of words of size $k+1$. We now compute $S_{j,k+1}$ where $j \leq k+1$. To this we add the optimal slack of the remaining sets of words. These sets can be found because they are of size less than $k+1$, and our inductive hypothesis assures this number is minimized. We then can figure out the minimum $S^2_{k,i} + opt[k-1]$. Thus, our algorithm works. □