1 Problem 3

Let slack $S_{j,k}$ be the difference

$$S_{j,k} = L - \left[\sum_{i=j}^{k-1}(c_i + 1)\right] + c_k \tag{1}$$

where if $S_{j,k} < 0$, $S_{j,k}$ = Integer.MAX_VALUE.

1.1 Algorithm

Segmented-slack(n)

1. Array $M[0...n]$
2. Set $M[0] = 0$
3. For all pairs $i \leq j$
   - Compute the slack $S_{i,j}$ for the line of words $w_i, ..., w_j$
4. Endfor
5. For $j = 1, ..., n$
   - Use the recurrence relation to compute $M[j]$
6. Endfor
7. Return M[n]

Find-lines(j)

1. If $j = 0$ then return
2. Else
   - Find an i that minimizes $S_{i,j}^2 + M[i-1]$
   - Output the line $w_i, ..., w_j$ and the result of Find-lines$(i-1)$
3. Endif

Recurrence Relation For the subproblem on the words $w_i, ..., w_j$,

$$OPT(j) = min_{1 \leq i \leq j}(S_{i,j}^2 + OPT(i-1)),$$

and the line $w_i, ..., w_j$ is used in an optimum partitioning for the subproblem if and only if the minimum is obtained using index i.

1.2 Runtime

To compute all the values of the slacks $S_{i,j}$, there are $O(n^2)$ pairs (i, j) for which this computation is needed. For each pair (i, j), we can use equation (1) to compute $S_{i,j}$ in $O(1)$ time. Thus, the total running time to compute all $S_{i,j}$ values is $O(n^2)$. Iterating through array M to find the optimal sequence of partitions to get value M[n] takes $O(n)$ time. Thus, the total running time to compute the partitions is bounded by $O(n^2)$.

1.3 Correctness

Lemma 1. *Segmented-slack(j) writes OPT(j) into the array entry M[j]*

Proof of Lemma 1. By definition $OPT(0) = 0$. Now, for some $j > 0$, assume that Segmented-slack(i) correctly computes $OPT(i)$ for all $i < j$. By the induction hypothesis,

$$\begin{aligned}\text{Segmented-slack}(j) &= min_{1\leq i\leq j}(S_{i,j}^2 + \text{Segmented-slack}k(i-1))\\ &= min_{1\leq i\leq j}(S_{i,j}^2 + OPT(i-1))\\ &= OPT(j).\end{aligned}$$

Hence, we have proved Lemma 1 as needed.