

Homework 8

Adam Karl

October 13, 2020

1 Problem 24 (4 points)

1.1 Setup

Create A, as a k by n matrix. A[i,j] will contain information on the optimal way to partition chapters x_1 through x_j into i volumes.

Each element in A will be a tuple (number of pages in the shortest volume, number of pages in the longest volume). Note that there may be multiple tuples, as (90, 100) and (95, 105) both have a range of 10 pages, but (90, 100) would be better to use if we add another volume with 90 pages, and (95, 105) would be better to use if we add another volume with 110 pages.

Create B, a 1D matrix with n elements. B holds information on possible final volumes. B[i] has the total number of pages for x_i through x_n . For this array, k is implied to be 1.

1.2 Algorithm Description

Fill in B. this is the pages in the last book starting at x_i , so it's just $x_i + \dots + x_n$ pages

- for i from 1 to n:
 - store the sum of $x_i + \dots + x_n$ in B[i]

Now fill in the first row of A. For this row k=1, so just one large volume with $x_1 + \dots + x_i$ pages

- for i in range 1 to n:
 - store the sum of $x_1 + \dots + x_i$ in A[1,i]

Fill in the rest of A by considering optimal solutions for the first p chapters with 1 fewer volume, combined with the volume made up from the remaining chapters.

- For i from 1 to k:
 - for j from 1 to n:
 - * //looking for the best way to fill in A[i,j]
 - * for p from 1 to j:
 - //p represents the last chapter before the final book starts
 - Consider A[i-1, p] and B[p+1]. I
 - newMin = MIN(min from A[i-1, p] tuple, B[p+1])
 - newMax = MAX(max from A[i-1, p] tuple, B[p+1])
 - if newMin-newMax is less than the range of the tuple already in A[i,j], replace it
 - if newMin-newMax is equal to the range of the tuple already in A[i,j], but has a different minimum, add the tuple to the list
 - If we are trying to fill in A[5, 100], we'd look at each possible chapter p that the 4th volume ended at, and combine the optimal solution in A[4, p] to the final volume which has B[p+1] pages
 - we have to do this for each tuple in A[i-1,p]. Usually there's just one but sometimes it's possible to have 2 different ranges

1.3 Conclusion and Trace-back

The optimal range(s) (the minimum and maximum page numbers for each volume) are stored in $A[k,n]$. If there is more than 1, we may choose a solution arbitrarily.

To trace back the solution, we have to consider each place p the $k-1$ th volume might end on, and see if combining the solutions in $A[k-1, p]$ and $B[p+1]$ result in the range stored in $A[k,n]$ (the same way as in the critical loop of the algorithm). When we find the place where this happens, chapters x_{p+1} through x_n are in the final volume.

We may repeat this step to determine which volume every chapter belongs to.