University of Toronto

CSCB70S—Data Structures and Techniques, Spring 1999

Assignment 4: Dynamic programming

Due: Wednesday April 7 at 10:00 am.
Worth: 10%.

The problem

Suppose we have an array divided into N segments, say $S_1, S_2, \ldots, S_N$ of sizes $\ell_1, \ell_2, \ldots, \ell_N$. Each of the segments is sorted. In order to sort the array, we may merge adjacent segments forming a larger sorted segment. The cost of merging two segments is measured as the number of comparisons employed, i.e., for segments of lengths r and s the cost is $r + s - 1$. We want to determine what is the best order to combine the segments, so as to have minimum total cost.

For example, if $N = 4$ and we have four segments S_1, S_2, S_3, S_4 with lengths $\ell_1 = 13$, $\ell_2 = 1$, $\ell_3 = 5$, and $\ell_4 = 2$, then we may combine the segments in the following ways (where the symbol "$\oplus$" represents a merge operation):

Merging sequence:	Total Cost:
$(((S_1 \oplus S_2) \oplus S_3) \oplus S_4)$	$13 + 18 + 20 = 51$
$((S_1 \oplus S_2) \oplus (S_3 \oplus S_4))$	$13 + 6 + 20 = 39$
$((S_1 \oplus (S_2 \oplus S_3)) \oplus S_4)$	$5 + 18 + 20 = 43$
$(S_1 \oplus ((S_2 \oplus S_3) \oplus S_4))$	$5 + 7 + 20 = 32$
$(S_1 \oplus (S_2 \oplus (S_3 \oplus S_4)))$	$6 + 7 + 20 = 33$

So in this example, the best way of merging the segments is as in the 4th merging sequence above, with a total cost of 32 comparisons.

The program

Let $C_{i,j}$ be the minimum cost of combining the segments $S_i, S_{i+1}, \ldots, S_{j-1}, S_j$. First, write a recurrence relation for $C_{i,j}$ for any $1 \le i, j \le N$.
Hint: for some k, assume that you know $C_{i,j}$ for all i and j such that $j - i = k$ and write $C_{i,j}$ for all i and j such that $j - i = k + 1$ in terms of those values. Make sure to include enough base cases and to justify your answer.
Then, get a copy of the starter code from the course webpage, and write a dynamic programming function in C++ to calculate the minimum total cost of combining N given segments of an array. You should add all your code to the one file in the starter code (since there is not that much to do in this assignment, all your code should be in one file).

Read the comments in the skeleton file **carefully**!

The comments will describe clearly where you need to add or change code, and the output format you must follow. The output must be the best order in which to combine the segments and the minimum total cost. As usual, write good, clear documentation for all your code, and make sure that you use the command-line options `-ansi -pedantic -Wall -g` when compiling the program.

Manual computations

Suppose we want to combine the array of digits `1240875396` from six segments: $S_1, S_2, S_3, S_4, S_5, S_6$. Note that the elements of the array are digits from 0 to 9 (the array has 10 positions). The segments have the following lengths and contain the following digits:

segment	S_1	S_2	S_3	S_4	S_5	S_6
length	3	2	1	1	2	1
numbers	`124`	`08`	`7`	`5`	`39`	`6`

Use dynamic programming **by hand** to find the sequence of combinations that has minimal total cost. For this, you will fill in two tables with your intermediate results:

- in the first table, the element at row r, column c contains the minimum total cost of combining the segments $S_r, S_{r+1}, \ldots, S_c$;
- in the second table, the element at row r, column c contains the index of the segment between S_r and S_c after which the *last* (or *outermost*) merge operation should be done (for example, for the combination $((S_3 \oplus (S_4 \oplus S_5)) \oplus S_6)$, the last merge operation happens after segment number 5, so the entry at row 3, column 6 would be "5").

Once you have filled in the two tables, give the cost and the order of combination of segments for the optimal sequence of combinations.

What to hand in

Test your program with the example above and with another example of your choice. Submit your program electronically using the following command from the directory that contains your code:

```
% submit -N a4 cscb70s *
```

Also hand in a copy of your source code, the output from the two tests, your recurrence relation for $C_{i,j}$, and your manual solution. Make sure that you justify your recurrence relation (explain how you came up with it and why you think it is correct), and show all your work for the manual computation (including all the values that you tested in determining each table element).

As usual, put your assignment in an envelope and attach the completed cover page to the front of the envelope.

COVER SHEET FOR ASSIGNMENT 4

Account name:

Surname:

Given Name(s):

Student Number:

Tutorial:

I declare that this assignment is my own work and is submitted in accordance with the University of Toronto Code of Behavior on Academic Matters, the Code of Student Conduct, and the guidelines for avoiding plagiarism described in the course syllabus.

I discussed this assignment with the following people:

Signature

GRADING

Program		
Correctness	________	/ 15
Recurrence	________	/ 5
Output	________	/ 5
Program style, including comments	________	/ 10
Manual computations		
Correctness	________	/ 15
Total	________	/ 50