Group Assignment 3: Divide & Conquer

CS325 Fall 2015

Due: Tuesday, November 17 at 10AM

You are encouraged to work in groups of up to three students. Only one member of each group should submit the group's work to TEACH, including the ***project report as a pdf*** *and the* **code** *you wrote. The report should have* ***all member's names included****. You may use any language you choose for implementation.* ***No questions about this assignment will be answered after the due date above.***[1]

The Most Magical Subsequence

Consider the following problem: Given an array of n non-negative integers $A[0, 1, \ldots, n-1]$, find indices i and j such that $i \leq j$ that maximizes:

$$\left(\min_{k=i,\ldots,j} A[k]\right) \times \left(\sum_{k=i}^{j} A[k]\right)$$

That is, given an array, find a subarray whose sum multiplied by the minimum number in that subarray is maximized. We call the answer the *most magical subsequence*. For example, if

$$A = [31, 41, 59, 26, 53, 58, 6, 97, 93, 23]$$

then the indices $i = 7$ and $j = 8$ has value

$$(\min\{97, 93\}]) \times (97 + 93) = 17670$$

and this is the best you can do.

You can solve this problem by enumerating over all $\Theta(n^2)$ choices for i and j and computing the sum and the minimum value, keeping the best *magical subsequence* found so far. By reusing computation as you did for the maximum subarray problem in the first group assignment, this algorithm will take $\Theta(n^2)$ time. In this assignment, you will develop a divide-and-conquer algorithm that is asymptotically faster. The divide-and-conquer algorithm requires a subroutine to compute the *most magical subsequence* that contains a particular element.

Greedy subroutine

Suppose you wish to find the *most magical subsequence* that contains $A[i]$ for a particular i. For example if

$$A = [31, 41, 59, 26, 53, 58, 6, 97, 93, 23]$$

and $i = 1$ then the *most magical subsequence* that contains $A[1] = 41$ is: [31, 41, 59, 26, 53, 58] and its corresponding value is $(31 + 41 + 59 + 26 + 53 + 58) \times 26 = 6968$.

[1]The due date in TEACH is 24 hours late, as submissions submitted within 24 hours of the deadline above will not be penalized.

One can find this greedily by generating a set of $n-1$ possible subsequences (instead of $\Theta(n^2)$ possible subseqences) of A and taking the best of these. Starting with $A[i]$, one considers $A[i, i+1]$ if $A[i+1] \geq A[i-1]$ and $A[i-1, i]$ if $A[i-1] \geq A[i+1]$, breaking ties *arbitrarily.* If one has added $A[j, \ldots, k]$ where $j \leq i \leq k$ to the set of possible subsequences, then one adds $A[j-1, \ldots, k]$ or $A[j, \ldots, k+1]$ based on whether $A[j-1]$ or $A[k+1]$ is larger. After generating this set of $n-1$ possible subsequences, one returns the subsequence with maximum value. Note that you should very efficiently compute the value of $A[j-1, \ldots, k]$ from the value of $A[j, \ldots, k]$.

For the above example, the set of possible subsequences of $A = [31, 41, 59, 26, 53, 58, 6, 97, 93, 23]$ that contains $A[1]$ that you would generate is:

$[41]$	$41 \times (41) = 1681$
$[41, 59]$	$41 \times (41 + 59) = 4100$
$[31, 41, 59]$	$31 \times (31 + 41 + 59) = 4061$
$[31, 41, 59, 26]$	$26 \times (31 + 41 + 59 + 26) = 4082$
$[31, 41, 59, 26, 53]$	$26 \times (31 + 41 + 59 + 26 + 53) = 5460$
$[31, 41, 59, 26, 53, 58]$	$26 \times (31 + 41 + 59 + 26 + 53 + 58) = \mathbf{6968}$
$[31, 41, 59, 26, 53, 58, 6]$	$6 \times (31 + 41 + 59 + 26 + 53 + 58 + 6) = 1644$
$[31, 41, 59, 26, 53, 58, 6, 97]$	$6 \times (31 + 41 + 59 + 26 + 53 + 58 + 6 + 97) = 2226$
$[31, 41, 59, 26, 53, 58, 6, 97, 93]$	$6 \times (31 + 41 + 59 + 26 + 53 + 58 + 6 + 97 + 93) = 2784$
$[31, 41, 59, 26, 53, 58, 6, 97, 93, 23]$	$6 \times (31 + 41 + 59 + 26 + 53 + 58 + 6 + 97 + 93 + 23) = 2922$

As part of your project, you will prove that this greedy approach does indeed find the *most magical subsequence* that contains $A[i]$.

A divide-and-conquer approach

If we choose the middle index m and split the input array into two halves, we know that the *most magical subsequence* will either be

- contained entirely in the first half $A[0, \ldots, m-1]$,
- contained entirely in the second half $A[m+1, \ldots, n-1]$, or
- made of a suffix of the first half, $A[m]$, and a prefix of the second half.

If we have computed these three options (the first two options recursively), then we simply take the best of the three options. For our example array A used above, we would take the best of

- the *most magical subsequence* of subarray of $[31, 41, 59, 26]$
- the *most magical subsequence* of subarray of $[58, 6, 97, 93, 23]$
- the *most magical subsequence* formed by a suffix of $[31, 41, 59, 26]$, $[53]$, and a prefix of $[58, 6, 97, 93, 23]$

To find the latter case, you should use the greedy algorithm in the previous section.

Tasks

Note: Use 64 bits data type to prevent overflow.

1. Write pseudocode for finding the *Most Magical Subsequence* problem described above that runs in $O(n^2)$ and give a full analysis its running time.
2. Write pseudocode for greedy algorithm that we described above that runs in $O(n)$ time and give a full analysis of its running time.

3. Prove that the greedy algorithm is correct: that it does indeed find the *most magical subsequence* that contains $A[i]$.

4. Write pseudocode for a divide and conquer algorithm for the *most magical subsequence* problem that uses the greedy algorithm as a subroutine and give a full analysis of its running time.

5. Implement the divide and conquer algorithm. A file containing test sets can be found here: `http://eecs.orst.edu/~glencora/cs325/magic/test_cases_with_solutions.txt` The file has 5 cases, each test case starts with a number n that represents the size of the array. in the second line there are n non-negative numbers. The solution to the test cases are afterward in a line that contains three separated number with spaces; most magical sequence and the corresponding start and end indices (with indices starting at 0). A line corresponding to the example above would be:

   Sample Input:
   `10`
   `31 41 59 26 53 58 6 97 93 23`
   Sample Output:
   17670 7 8

   You may use this test file to check that your code is correct. You should also test your code on small hand-generated instances. **Submit your code in a separate file to TEACH.**

6. Another folder containing 8 instances (T1.txt through T8.txt) without solutions can be found here: `http://eecs.orst.edu/~glencora/cs325/magic/` All files starts with n (the size of the array) in the first list, in the second line there are n non-negative space-separated numbers. **Submit a text file to TEACH named `answers.txt` that contains your answers for the test cases.** Each line of your file represents your answer for the test case on the corresponding name of test cases file that is solution to T1 should be on the first line and solution to T8.txt should be on the 8th line. This means that your file should contain exactly 8 lines. Each line of your file should contain three numbers separated by white space (spaces or tabs). The first number should be the value of *most magical subsequence*, the second number should be the start index, and the last number should be the end index; note that indices start at 0.