Learning outcomes

The purpose of this exercise is for you to demonstrate the following learning outcomes and for me to assess your achievement of them.

1. To demonstrate how the study of algorithmics has been applied in a number of different domains.

2. To introduce formal concepts of measures of complexity and algorithms analysis.

3. To introduce fundamental methods in data structures and algorithms design.

Problem Description

Suppose that we are given an array $A[1,2,...,n]$ containing $n \geq 3$ positive, not necessary distinct, integers. We do not assume that the input array is sorted. We first define a notion of a *peak*. Any three consecutive indices $i,i+1,i+2$ such that $1 \leq i \leq n-2$ and $A[i] < A[i+1]$ and $A[i+1] > A[i+2]$, is called a *peak*; note that the inequalities are strict.

A *range* is a collection of disjoint peaks. Formally, a *range* that consists of $k \geq 1$ peaks is the following collection of indices in array A: $\{i_1,i_2,...,i_k\}$ such that:

(1) $1 \leq i_1 < i_2 < \cdots < i_k \leq n-2$ (indices are distinct increasing integers from $\{1,2,...,n\}$),

(2) $i_1+2 < i_2, i_2+2 < i_3, \cdots, i_{k-1}+2 < i_k$ (peaks are disjoint, i.e., indices are separated from one another by at least 2 positions in array A),

(3) Each three consecutive indices i_j,i_j+1,i_j+2 (for $j = 1,2,...,k$) form a peak, that is, $A[i_j] < A[i_j+1]$ and $A[i_j+1] > A[i_j+2]$ for each $j = 1,2,...,k$ (we have k disjoint peaks)

(4) For every two consecutive peaks i_j,i_j+1,i_j+2 and $i_{j+1},i_{j+1}+1,i_{j+1}+2$, (for $j = 1,2,...,k-1$), we have the following additional condition:
$A[i_j+1] \geq A[i_{j+1}]$ and $A[i_j+2] \leq A[i_{j+1}]$ (note weak inequalities here). This condition intuitively means that the east slope of the peak i_j,i_j+1,i_j+2 contains the base (first element i_{j+1}) of the west slope of the very next peak $i_{j+1},i_{j+1}+1,i_{j+1}+2$.

If a *range* consists of $k \geq 1$ peaks, then its *length* is k. The Longest Range of Peaks Problem is to find the length of the longest range of peaks in the input array A, or output 0 if there is no peak in array A.

Examples:

Let us consider input $A[1,6,2,11,2,10,5,7,3]$ with n = 9. Here $A[1]$,$A[2]$,$A[3]$; $A[3]$,$A[4]$,$A[5]$; $A[5]$,$A[6]$,$A[7]$; $A[7]$,$A[8]$,$A[9]$ are four peaks in A. These are all peaks in array A. We also have the following range of length 2: $A[1]$,$A[2]$,$A[3]$; $A[5]$,$A[6]$,$A[7]$. Another range of length 2 is: $A[3]$,$A[4]$,$A[5]$; $A[7]$,$A[8]$,$A[9]$. And another range of length 2 is: $A[1]$,$A[2]$,$A[3]$; $A[7]$,$A[8]$,$A[9]$. Here, any of these 3 ranges is the the longest range in this instance of the problem and so the output of the Longest Range of Peaks Problem is 2.

Let us consider now input $A[1,6,2,2,2,10,5,7,8,3]$ with n = 10. Here, the longest range has length 3 and it is: $A[1]$,$A[2]$,$A[3]$; $A[5]$,$A[6]$,$A[7]$; $A[8]$,$A[9]$,$A[10]$, so the output of the Longest Range of Peaks Problem is 3. Observe, for instance, that $A[1]$,$A[2]$,$A[3]$; $A[8]$,$A[9]$,$A[10]$ is **not** a range because the condition (4) above is false, namely, $A[8]$ = 7 is not between $A[3]$ = 2 and $A[2]$ = 6.

If the input array is sorted in non-decreasing order, for instance $A[1,6,8,8,10,13]$, then there is no peak, and so the output of the Longest Range of Peaks Problem is 0. Similarly, if the array is sorted in non-increasing order, for instance $A[15,11,4,4,3,3]$, then there is no peak, and so the output of the problem is 0. Some further examples of inputs.

Suppose, for instance, that n = 13 and that the input sequence is:

1 3 2 1 7 5 7 10 1 1 1 3 2,

that is, $A[1]$ = 1,$A[2]$ = 3,$A[3]$ = 2,$A[4]$ = 1,$A[5]$ = 7,$A[6]$ = 5,$A[7]$ = 7,$A[8]$ = 10,$A[9]$ = 1,$A[10]$ = 1,$A[11]$ = 1,$A[12]$ = 3,$A[13]$ = 2. Then, the longest range of peaks is: $A[4]$,$A[5]$,$A[6]$; $A[7]$,$A[8]$,$A[9]$; $A[11]$,$A[12]$,$A[13]$ and has length 3. The output to the problem is 3.

Suppose, for instance, that n = 11 and that the input sequence is:

1 5 2 2 2 7 4 4 4 5 4,

that is, $A[1]$ = 1,$A[2]$ = 5,$A[3]$ = 2,$A[4]$ = 2,$A[5]$ = 2,$A[6]$ = 7,$A[7]$ = 4,$A[8]$ = 4,$A[9]$ = 4,$A[10]$ = 5,$A[11]$ = 4. Then, the longest range of peaks is: $A[1]$,$A[2]$,$A[3]$; $A[5]$,$A[6]$,$A[7]$; $A[9]$,$A[10]$,$A[11]$ and has length 3. The output to the problem is 3.

For further examples of inputs together with answers, see the text file dataTwo.txt that I provide (see explanation of the data format below). The file dataTwo.txt contains also solutions.

The task: You should design a dynamic programming algorithm and write a procedure to implement the sequential implementation of your dynamic programming algorithm, that for any given input sequence of *n* positive integers (multiple identical numbers allowed) finds the length of the longest range of peaks (or 0 if there is no peak in the sequence). Your procedure should only output the value of the longest range of peaks or 0.

Additionally, you should include a brief idea of your solution in the *commented* text in your code, describing how you derive your recursive dynamic programming solution first and ideas of its sequential implementation. You should also include a short analysis and justification of the running time of your procedure in terms of *n*. These descriptions are part of the assessment of your solution.

Description of the input data format:

Input data text file dataOne.txt has the following format (this example has 2 inputs, each input ends with A; note the number 0 is the part of the input format, but not part of the input sequences):

$$\begin{array}{c} 0 \\ a_1 \\ a_2 \\ \cdots \\ \cdots \\ a_n \\ A \\ 0 \\ a_1 \\ a_2 \\ \cdots \\ \cdots \\ a_n \\ A \end{array}$$

In general file dataOne.txt can have an arbitrary number of distinct inputs of arbitrary, varying lengths. File dataOne.txt contains 50 different instances of the problem. The first 6 instances are the same as the examples above. Observe that *n* is not explicitly given as part of the instance. Also 0 which starts each instance does not have any particular purpose; it is just format of the input data to mark beginning of an instance.

Input data text file dataTwo.txt has the following format (this example has again 2 inputs, each input ends with A):

$$
\begin{gathered}
0 \\
ans_1 \\
a_1 \\
a_2 \\
\dots \\
\dots \\
a_n \\
A \\
0 \\
ans_2 \\
a_1 \\
a_2 \\
\dots \\
\dots \\
a_n \\
A
\end{gathered}
$$

There ans_1 (ans_2, respectively) is the value of the longest range for the first (second, respectively) instance or 0 if there is no peak in those instances. File dataTwo.txt contains the same 50 instances of the problem as in file dataOne.txt but in addition has the answers. This data can be used to test the correctness of your procedure.

Again, in general file dataTwo.txt can have an arbitrary number of distinct input sequences of arbitrary, varying lengths. It is provided by me with correct answers to these instances.

The solutions shown in dataTwo.txt are (at least) the claimed solutions for each sample input, computed by my program. Recall that your solution should print out the length of the longest range in the given sequence for the given instance, or 0 in case if this instance contains no peaks. Note, that your program does not need to output this longest range of peaks.