Question 1. [12 marks]

Given an array of $n \geq 2$ integers, say $[x(1), \ldots, x(n)]$, we want to find the largest step $d$, which is defined to be the max of $x(j) - x(i)$ over all $j > i$. For example, for $x = [22, 5, 8, 10, -3, 1]$

\[ d = x(4) - x(2) = 10 - 5 = 5. \]

Part (a) [5 marks]

Finish the pseudo-code $LS$ below for computing $d$. To get any marks at all your algorithm must make essential use of a divide and conquer strategy which splits the current problem roughly in half. (We realize there are simpler ways to compute $d$.)

```plaintext
// For the function defined further below, the largest step $d$ is given by,
[d] = LS(x, 1, n)
// Assume a function call can return more values than are actually assigned to variables on the left.

[d, ________________] = LS(x, a, b)
// Input: Array x(1..n) of integers, and indicies 1 $\leq a < b \leq n$.
// We require $n \geq 2$.
// Output: Largest step $d = x(j) - x(i)$, for any $i, j$ with $a \leq i < j \leq b$
// Feel free to add additional returned values or arrays, as needed. To do that
// use “return [r1, r2, r3, ...]” to return more than one value in your pseudo-code.
```

Total pages = 5 Page 2 CONT’D...
Question 1.  (continued)

Part (b)  [2 marks]

Given an array $x$ of length $n$, let $T(n)$ be the runtime of $LS(x, 1, n)$ in part (a). Write an equation expressing $T(n)$ in terms of $T(k)$ for some integer(s) $k < n$. Briefly explain.

Part (c)  [1 mark]

In parts (c) and (d) we design a dynamic programming solution for the largest step $d$. To begin, give a simple expression for $LS(x, 1, 2)$ in terms of $x$.

Part (d)  [4 marks]

Finally, write an equation for $LS(x, 1, k + 1)$ in terms of $LS(x, 1, k)$ for values $2 \leq k < n$. (Your equation may use other simple properties of the array $x$, and it must make essential use of $LS(x, 1, k)$.) Briefly explain why your equation is correct.
Question 2. [12 marks]

Given an array of \( n \) integers \([x(1), \ldots, x(n)]\) define an increasing subsequence of \( x \), to be a subsequence \( x(j(1)) < x(j(2)) < \ldots < x(j(p)) \) where the indices \( j(i) \) are also strictly increasing with \( i \), that is, \( 1 \leq j(1) < j(2) < \ldots < j(p) \leq n \). We are interested in a longest increasing subsequence of \( x \), i.e., with the maximum length \( p \). Define \( LLIS(x) \) to be the length of a longest increasing subsequence of \( x \) (there may be more than one increasing subsequence with this length).

For example, given the array \( x = [22, 5, 8, -3, 10, 1] \) the longest increasing subsequence is \([5, 8, 10]\) which has length is 3. Therefore, for this example, \( LLIS(x) = 3 \).

In addition, define the array \([q(1), \ldots, q(n)]\) to have elements \( q(k) \) which equal the length of the longest increasing subsequence of \( x \) ending with the value \( x(k) \). In particular, such a subsequence only uses values from the first \( k \) elements of \( x \), namely \([x(1), \ldots, x(k)]\) and must end with \( x(k) \) itself.

For the example array \( x \) above, \([q(1), \ldots, q(6)] = [1, 1, 2, 1, 3, 2] \).

Part (a) [5 marks]

Give an equation which expresses \( q(k) \) for \( 1 < k \leq n \) in terms of the values of \( q(j) \) for various \( j < k \) (and possibly other simple expressions involving elements of the array \( x \)). Explain.

Part (b) [2 marks]

Express \( LLIS(x) \) in terms of the array \( q \). Explain.
Question 2.  (continued)

Part (c)  [5 marks]

Given the arrays $x$ and $q$ defined above, provide pseudo-code for an algorithm which extracts a longest increasing subsequence. (Precisely worded English sentences describing simple steps are allowed in your pseudo-code.) In the previous notation, the algorithm should return $[x(j(1)), x(j(2)), \ldots, x(j(p))]$ for the maximum $p$. The $j(k)$’s themselves do not need to be returned, just the corresponding $x$ values (in order). Briefly explain why your algorithm is correct.