Answer all questions. Give informal (at least) proofs for all answers. Grading will be on completeness and logical correctness, and if applicable, efficiency, as well as correctness. Out of 80 points.

Order questions- 10 points each For each, answer True or False, and give a short (1-2 sentence) explanation for your answer.

1. \( n + (n - 1) + (n - 2) \in O(n) \)
2. \( n + (n - 1) + (n - 2) + (n - 3) + \ldots + 2 + 1 \in O(n) \).
3. If \( f \) and \( g \) are functions from positive integers to positive integers, and \( f(n) \in O(g(n)) \), then \( f(n) \cdot g(n) \in O((g(n))^2) \). (If not always true, give an example of functions \( f \) and \( g \) for which it is false.)
4. If \( f \) is functions from positive integers to positive integers, then \( f(2n) \in O(f(n)) \). (If not always true, give an example of a function \( f \) for which it is false.)

Analyzing loops-20pts Here is an algorithm that, given an array of integers \( A[1] \ldots A[n] \), outputs an array \( B[1..n] \), where for each \( I \), \( B[I] \) is either the first \( J \) greater than \( I \) with \( A[J] > A[I] \) or \( n + 1 \) if no such \( J \) exists. The algorithm uses a stack of pairs \((position, value)\), \( Stack \), with constant time operations \( Top \) (returning either the top element or a “stack empty” message if the stack is empty) \( Push \) and \( Pop \). \( Infinity \) is a value bigger than any array position. Give a time analysis for the algorithm, in \( O \) notation.

Intersect \( (A[1..n]: \text{sorted list of positive integers}) \)

1. Initialize an empty stack \( Stack \) of pairs with two fields, position and value
2. \( Push(n + 1, Infinity) \)
3. FOR \( I = n \) down to 1 do:
   4. While \( A[I] \geq Top.value \) do \( Pop \)
5. \( B[I] \leftarrow Top.position \)
7. Return \( B \).

Correctness Proofs: 20 points Prove the following loop invariant about the algorithm above, that is useful in showing the algorithm correct.

Lemma: After each iteration, both the positions and values of the elements in the stack are increasing from top to bottom.