Algorithms

Assignment: Binary Search

Name: 

Id: 

Grade

Good Luck!
1. Assume \( n \geq 1 \) is a power of 2. Let \( A = [A_1 \leq A_2 \leq \cdots \leq A_n] \) be a sorted array of \( n \) integers. Let \( x \) be an integer.

Design an efficient algorithm that finds the number of times \( x \) appears in the array. What is the worst-case number of comparisons made by your algorithm?

Justify the correctness and complexity of your algorithm.
2. Assume \( n \geq 1 \) is a power of 2. Let \( A = [A_1 < A_2 < \cdots < A_n] \) be a sorted array of \( n \) distinct positive integers. Let \( x \leq y \) be two positive integers.

Describe an efficient algorithm that determines if one of the numbers \( x, x+1, \ldots, y \) appears in the array. What is the worst-case number of comparisons made by your algorithm?

Justify the correctness and complexity of your algorithm.
3. Assume \( n \geq 1 \) is a power of 2. Let \( A = [A_1 < A_2 < \cdots < A_n] \) be a sorted array of \( n \) distinct positive integers. Let \( x \leq y \) be two positive integers.

Describe an efficient algorithm that determines if \textbf{all} the numbers \( x, x + 1, \ldots, y \) appear in the array. What is the worst-case number of comparisons made by your algorithm?

Justify the correctness and complexity of your algorithm.
4. Let $A = [A_1 < A_2 < \cdots < A_n]$ be a sorted array of $n$ distinct integers (could be negative and/or negative).

Describe an efficient algorithm that finds, if exists, an index $1 \leq i \leq n$ such that $A_i = i$. What is the worst-case number of comparisons made by your algorithm?

Justify the correctness and complexity of your algorithm.
5. Let $A = [A_1 < A_2 < \cdots < A_n]$ be an array of $n$ distinct integers sorted in an ascending order and let $B = [B_1 > B_2 > \cdots > B_n]$ be an array of $n$ distinct integers sorted in a descending order.

Describe an efficient algorithm that finds, if exists, an index $1 \leq i \leq n$ such that $A_i = B_i$. What is the worst-case number of comparisons made by your algorithm? Justify the correctness and complexity of your algorithm.