Consider the following Java code:

```java
boolean isSorted(int[] a) {
  return isSortedHelper(a, 0, a.length - 1);
}
boolean isSortedHelper(int[] a, int from, int to) {
  if (from > to) {
    return true;
  }
  else if(from == to) {
    return true;
  }
  else {
    return a[from] <= a[from + 1]
      && isSortedHelper(a, from + 1, to);
  }
}
```

Prove, via mathematical induction, that the method **isSorted** method above correctly returns `true` if the array `a` is sorted in a non-descending order; and `false` otherwise.

**Solution**:

We first prove that the recursive helper method **isSortedHelper** (Line 4 – Line 15) is correct (i.e., is the subarray $\{$`a[from]`, `a[from + 1]`,..., `a[to]`$\}$ sorted).

1. **Base Cases**

   (a) [**4 marks**] **Concept**: In an empty array, there is no witness (i.e., adjacent numbers that are not sorted) $\therefore$ result is **true**.

   (b) [**4 marks**] **Link to Code**: **Lines 5 − 7 (or just Line 6)** of the above code does this.

   (c) [**4 marks**] **Concept**: In an array of size 1, the only one element is automatically sorted.

   (d) [**4 marks**] **Link to Code**: **Lines 8 − 10 (or just Line 9)** of the above code does this.

2. **Inductive Cases**

   (a) [**4 marks**] **Inductive Hypothesis (I.H.)**: The recursive call `isSortedHelper(a, from + 1, to)` returns **true** if a[from + 1], a[from + 2], ..., a[to] are sorted in a non-descending order; **false** otherwise.

   (b) [**4 marks**] **Concept**: `isSortedHelper(a, from, to)` should return **true** if:

   **1)** `a[from]` $\leq$ `a[from + 1]`; and

   **2)** the subarray $\{$`a[from + 1]`,..., `a[to]`$\}$ is sorted.

   (c) [**4 marks**] **Link to I.H.**: By I.H., condition **2)** is satisfied.

   (d) [**4 marks**] **Link to Code**: **Line 12** in the above code does condition **1)**.

   $\therefore$ **Lines 12 − Line 13** perform a correct combination.

3. Given that the recursive helper method **isSortedHelper** (Line 4 – Line 4) is correct, we now argue that the method **isSorted** (Line 1 – Line 3) is correct.

   (a) [**4 marks**] **Concept**: `isSorted(a)` is correct by invoking `isSortedHelper(a, 0, a.length - 1)`, examining the entire array.

   (b) [**4 marks**] **Link to Code**: **Line 2** of the above code does this.