

Correctness and the loop invariant

HackerRank Prepare > Algorithms > Sorting > Correctness and the Loop Invariant

Problem

Submissions

Leaderboard

Discussions

Loop Invariant
In the beginning, only the first element is "sorted".

Each element is sorted and placed in its currently "correct" spot.

Each turn, the subarray of sorted elements expands.

...Until the entire array is sorted.

```
23 System.out.print(n+" ");
24 }
25 }
26 public static void main(String[] args) {
27     Scanner in = new Scanner(System.in);
28     int n = in.nextInt();
29     int[] ar = new int[n];
30     for(int i=0; i<n; i++){
31         ar[i]=in.nextInt();
32     }
33     insertionSort(ar);
34 }
35 }
```

Line: 14 Col: 25

You have earned 30.00 points!
You are now 174.35 points away from the 4th star for your problem solving badge.

37% 300.65/475

Congratulations
You solved this challenge. Would you like to challenge your friends?

Test case 0

Compiler Message

You can often use a similar process to demonstrate the correctness of many algorithms. You can see [these notes](#) for more information.

Challenge

In the InsertionSort code below, there is an error. Can you fix it? Print the array

As talked about in class, the time complexity for this code would be $O(n^2)$ since it is the worst possible scenario for the code. It does not have a pointer element, and because of that the code starts at the far left of the Array. As well as that it also has a nested loop that makes the time complexity of insertion $O(n^2)$.

The Space complexity is a little different because it involves an input Array, and due to the input array the space complexity would be $O(n)$, even though the variables inside the Array have a constant space complexity, the Array itself has a space complexity of $O(n)$.