

Problem: Implement insertion sort.

- [Constraints](#)
- [Test Cases](#)
- [Algorithm](#)
- [Code](#)
- [Unit Test](#)

Constraints

- Is a naive solution sufficient?
 - Yes
- Are duplicates allowed?
 - Yes
- Can we assume the input is valid?
 - No
- Can we assume this fits memory?
 - Yes

Test Cases

- None -> Exception
- Empty input -> []
- One element -> [element]
- Two or more elements

Algorithm

6 5 3 1 8 7 2 4

Wikipedia's animation:

- For each value index 1 to $n - 1$
 - Compare with all elements to the left of the current value to determine new insertion point
 - Hold current value in temp variable
 - Shift elements from new insertion point right
 - Insert value in temp variable
 - Break

Complexity:

- Time: $O(n^2)$ average, worst. $O(1)$ best if input is already sorted
- Space: $O(1)$ for the iterative solution

Misc:

- In-place
- Stable

Insertion sort works well for very small datasets where most of the input is already sorted.