

Insertion Sort

General

Worst case: $O(n^2)$

Best case: $O(n)$

Average: $O(n^2)$

Pros: Efficient on small data sets

Stable = doesn't change order of elements with equal keys (relatively)

In-place = needs $O(1)$ additional memory space (ie. Uses the same array)

Online = able to sort a list as the unsorted list is received

Performs faster if a backwards linear search is used on partially sorted array

Cons: If using an array, insertion is slow because all elements after the selected element must be moved up

Description

During each iteration, an element (arbitrarily chosen) is removed from the input array and inserted into the correct position. Remember that the sorting is performed in-place.

A defined function – Insert – inserts a value into the proper location in a sorted list. Starting from the end of the sorted list and moving towards the beginning of the list, the selected value is compared against each value along the way. As a result of this operation, the stored value is immediately overwritten.

For arrays, the following procedure is typically followed:

1. Begin at start of array
2. Insert each element into proper location

Example

Original sequence: 64 25 12 22 11

64 25 12 22 11

- 64 is selected and moved into sorted array

25

___ 64 12 22 11

- 25 is selected & compared against 64

25 64 12 22 11

12

25 ___ 64 22 11

- 12 is selected & compared against 64

12

___ 25 64 22 11

- 12 is compared against 25

12 25 64 22 11

... process repeats for remaining #s

References

https://en.wikipedia.org/wiki/Insertion_sort

<http://www.brucemerry.org.za/manual/algorithms/sorting.html>