



Gnome Sort

Origin and Description

- Gnome sort is a sorting algorithm invented by Iranian computer scientist Hamid Sarbazi-Azad in 2000.
- Sarbazi-Azad originally called the sorting method “stupid sort” due to its simplicity, however Dutch computer scientist Dick Grune later gave it the name gnome sort.
- The name is based on the technique used by the standard Dutch Garden Gnome to sort a line of flower pots. Basically, he looks at the flower pot next to him and the previous one; if they are in the right order he steps one pot forward, otherwise, he swaps them and steps one pot backward. If there is no previous pot, he steps forwards; if there is no pot next to him, he is done.
- Gnome sort is a conceptually simple, stable comparison and exchange sort method like insertion sort in that it works with one item at a time but gets the item to the proper place by a series of swaps, like bubble sort.
- Due to its simplicity, it requires no nested loops and has a tiny code size yet is not very popular.



Complexity

- Best-case performance: $O(N)$
- Worst-case performance: $O(N^2)$
- Average-case performance: $O(N^2)$

Advantages



- Can sort in either direction (increasing or decreasing)
- Simple to implement and understand
- May run faster than non-comparison sorts on sorted or almost sorted lists

Disadvantages



- Slightly slower than bubble sort
- Average complexity is $O(n^2)$
- Not good on large data sets

Comparison

- With insertion sort: searches entire array for correct position while gnome sort uses pairwise swaps
- With bubble sort: compares two adjacent elements while gnome sort compares two adjacent elements then compares prior adjacent pairs

Optimization



- May be optimized by using a variable to store the position before traversing back toward the beginning of the list
- With this optimization, the gnome sort would become a variant of the insertion sort

Use

- Best suited for beginners who want to learn sorting
- Is not very efficient and not a good alternative to faster and/or longer sorting algorithms