

Mystery Sorters

About Me

I am Levi Butler, a passionate software developer interested in full stack development.

The Problem

There are 5 mysterious sorting algorithms, and it is up to me to figure out what method each algorithm uses. I know they can be one of the five algorithms

1. optimized bubble sorting
2. insertion sort
3. merge sort
4. quicksort (with last element as pivot)
5. selection sort

Strategy

I know that the algorithms have different time complexities so I can test each algorithm with different sizes of random arrays to match the time complexities of the mystery sorters. Also, I know some algorithms behave differently depending on if the original algorithm is already sorted, and even further distinction if it is sorted backwards. With these unique behaviors in mind, I planned to test all three parameters; random arrays of different sizes, ascending arrays of different sizes, and descending arrays of different sizes; and utilize the differences to identify each sorting algorithm.

Identified Algorithms

- mystery01: merge sort
- mystery02: optimized bubble sort
- mystery03: insertion sort
- mystery04: quick sort (with last element as pivot)
- mystery05: selection sort

Example Set

Sorter	Array Order	Size 10	Size 100	Size 1000
mystery01	Random	276	4187	55672
mystery01	Ascending	167	1889	19677
mystery01	Descending	169	1925	21336
mystery02	Random	179	12031	707809
mystery02	Ascending	31	90	644
mystery02	Descending	73	7376	640371
mystery03	Random	116	2424	162934
mystery03	Ascending	34	116	948
mystery03	Descending	48	3491	314120
mystery04	Random	169	2581	37211
mystery04	Ascending	75	3790	318233
mystery04	Descending	71	4388	389361
mystery05	Random	180	5879	352020
mystery05	Ascending	62	3616	315206
mystery05	Descending	53	3640	278024

Identifiers

- mystery01: merge sort
 - Fast for large sets
 - Still does many operation for sorted sets
- mystery02: optimized bubble sort
 - Extremely slow for random and backwards ordered sets
 - Extremely fast for correctly ordered sets, as the initial check for order will pass
 - Faster than insertion sort especially as sets grow
- mystery03: insertion sort
 - Second slowest for large unordered or backwards ordered sets
 - second fastest for large correctly ordered sets
- mystery04: quick sort (with last element as pivot)
 - Slowest with large in order sets as the pivot will be the worst case possible
 - Quickest with random sets
- mystery05: selection sort
 - Slowest across the board
 - Slow even when sets are ordered