Chapter 6 : Part 4

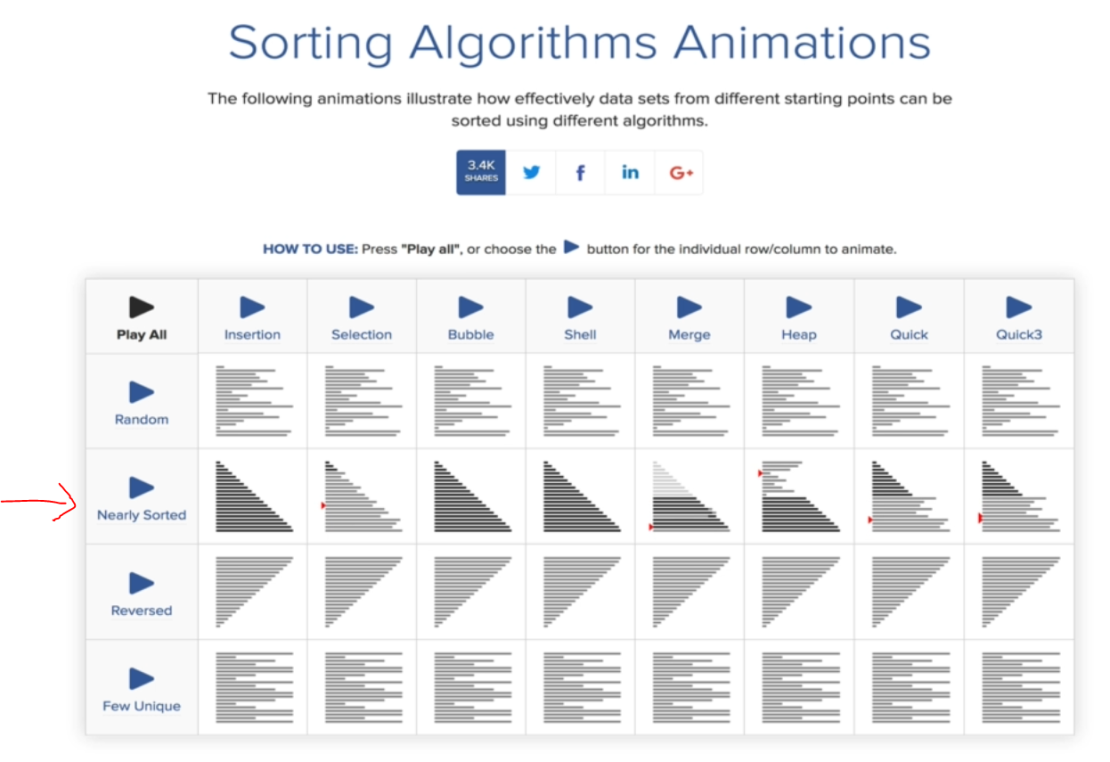
**Sorting Algorithms: Comparison**

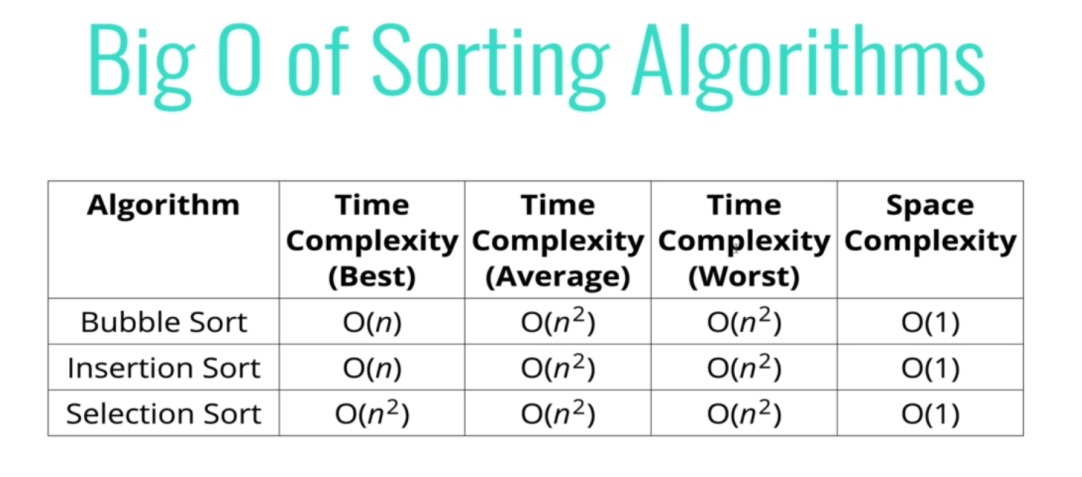
**Bubble vs Selection vs Insertion**

**6.4.1 Comparison**



* Nearly sorted data: *Bubble sort* (only loops twice) & *Insertion sort* (only loops once) works well.
* *Selection sort* however doesn't works so well in this case, it *loops n-times*, where *n* is the no. of *total array elements*.
* Quadratic: Note that, these three algorithms are Quadratic (because of their quadratic time complexity). And
* Small Data: These three algorithms only works well in smaller data. For bigger data we need much more complex sorting algorithms.





* Space Complexity: More or less these algorithms are *same for space complexity*.
* Because, we are not creating new spaces: such as, variables, arrays etc.
* Other algorithms, such as Marge Sort makes another smaller arrays to do sorting. So it has higher space complexity.
* Continuously coming Data (streamed data): Insertion sort specially used for these kind of data, where all previous data are *already* *sorted* and *new data continuously coming* and we have to place new data in *proper* *place*.

**Course Sections**

1. Introduction
2. Big O Notation
3. Analyzing Performance of Arrays and Objects
4. Problem Solving Approach
5. Problem Solving Patterns
6. 100% OPTIONAL Challenges
7. Recursion
8. Recursion Problem Set
9. Bonus CHALLENGING Recursion Problems
10. Searching Algorithms
11. Bubble Sort
12. Selection Sort
13. Insertion Sort
14. Comparing Bubble, Selection, and Insertion Sort
15. Merge Sort
16. Quick Sort
17. Radix Sort
18. Data Structures Introduction
19. Singly Linked Lists
20. Doubly Linked Lists
21. Stacks + Queues
22. Binary Search Trees
23. Tree Traversal
24. Binary Heaps
25. Hash Tables
26. Graphs
27. Graph Traversal
28. Dijkstra's Algorithm!
29. Dynamic Programming
30. The Wild West