**Sorts—A Review**

Bubble Sort

* short description of algorithm: compares side-by-side. Takes 2 nested loops.
* best case: O( n2 ) when the data is in order.
* worst case or guaranteed performance: O( n2 ) when the data is in reverse order.

Selection Sort

* algorithm: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* best case: O( ) when the data is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* worst case or guaranteed performance: O( ) when the data is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Insertion Sort

* algorithm: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* best case: O( ) when the data is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* worst case or guaranteed performance: O( ) when the data is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Merge Sort

* algorithm: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* average case or expected performance: O( ) when the data is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* worst case or guaranteed performance: O( ) when the data is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Quick Sort

* algorithm: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* average case or expected performance: O( ) when the data uses \_\_\_\_\_\_\_\_\_\_\_\_\_\_
* worst case or guaranteed performance: O( ) when the data uses \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Heap Sort

* algorithm: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* average case or expected performance: O( ) when the data is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* worst case or guaranteed performance: O( ) when the data is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Here are some sorting websites: <http://sorting.at/>

<https://www.cs.usfca.edu/~galles/visualization/ComparisonSort.html>

Wikipedia entries