Uy Nguyen

Experiment-1

Algorithms	***Input Type***	***#comparisons***	***Comments***
Heap S	random	258470	Valid as having O(NlogN)-(5-6 digits)
Merge S	random	120414	Valid as having O(NlogN)-(5-6 digits)
Quick S (fp)	random	106296	Valid as having O(NlogN)-(5-6 digits)
Quick S (rp)	random	112955	Valid as having O(NlogN)-(5-6 digits)
Selection S	random	49995000	Valid as having O(N^2)-(8-9 digits)

Experiment-2

Size of input		100	500	1000	5000	10,000
Algorithms	Selection S	4,950	124,750	499,500	12,497,500	49,995,000

Size of input		100	500	1000	5000	10,000
Algorithms	Merge S	538	3,874	8,664	55,229	120,336

Size of input		100	500	1000	5000	10,000
Algorithms	Quick S (fp)	481	3,232	7,254	47,775	112,479

Size of input		100	500	1000	5000	10,000
Algorithms	Quick S (rp)	620	3,347	7,184	50,844	111,197

Size of input		100	500	1000	5000	10,000
Algorithms	Heap S	1,264	8,544	19,182	119,286	258,204

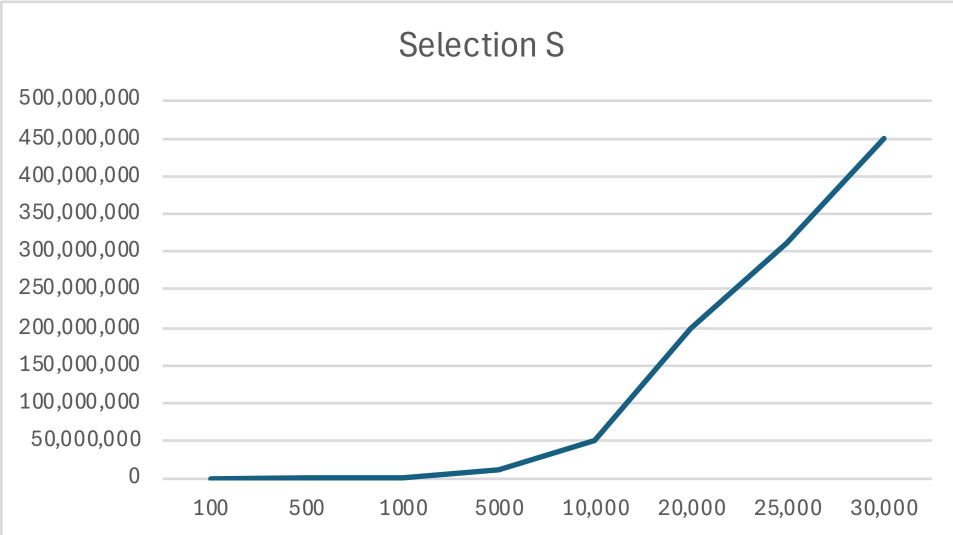
Size of input		20,000	25,000	30,000
Algorithms	Selection S	199,990,000	312,487,500	449,985,000

Size of input		20,000	25,000	30,000
Algorithms	Merge S	260,932	334,104	408,619

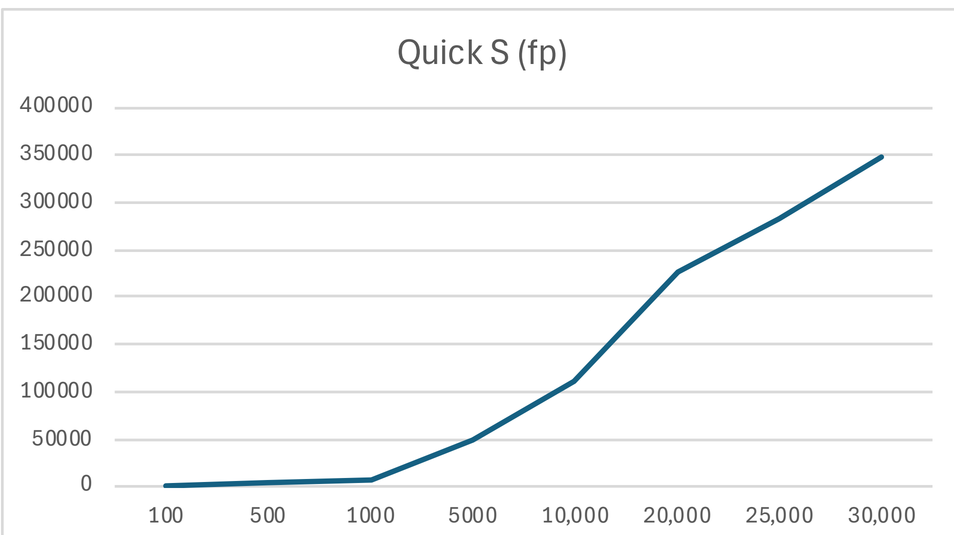
Size of input		20,000	25,000	30,000
Algorithms	Quick S (fp)	227,173	281,972	348,359

Size of input		20,000	25,000	30,000
Algorithms	Quick S (rp)	227,687	338,596	384,035

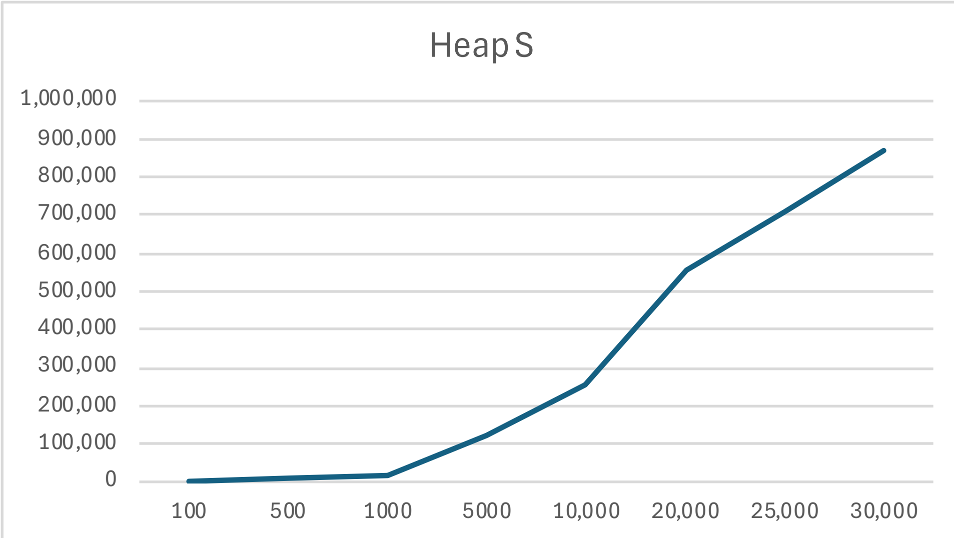
Size of input		20,000	25,000	30,000
Algorithms	Heap S	556,852	712,780	869,980



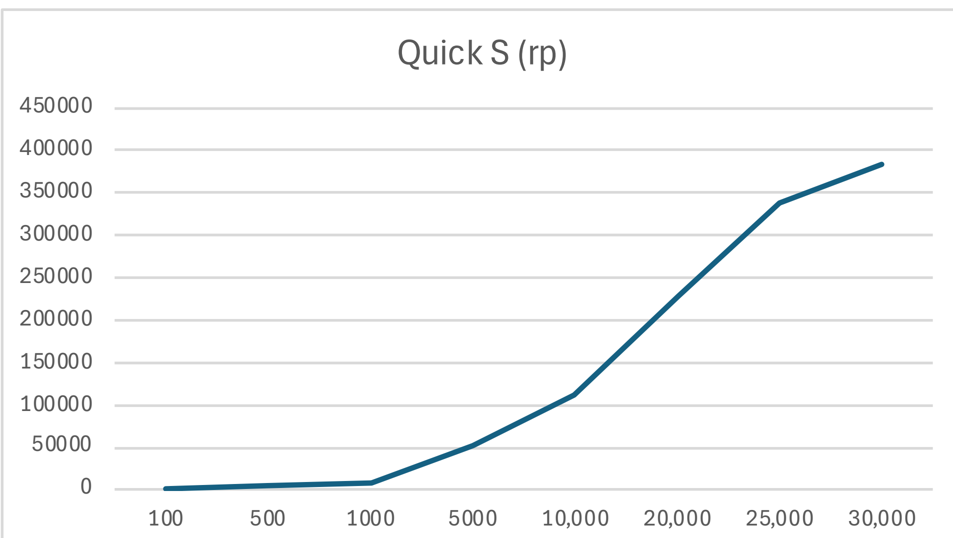
Selection Sort
Theoretical Time Complexity: $O(n^2)$
Analysis: The growth is quadratic, matching the theoretical $O(n^2)$.
Each 10× increase in input size results in ~100× increase in time,
which is characteristic of quadratic growth.
Conclusion: Experimental results perfectly match the theoretical expectation.



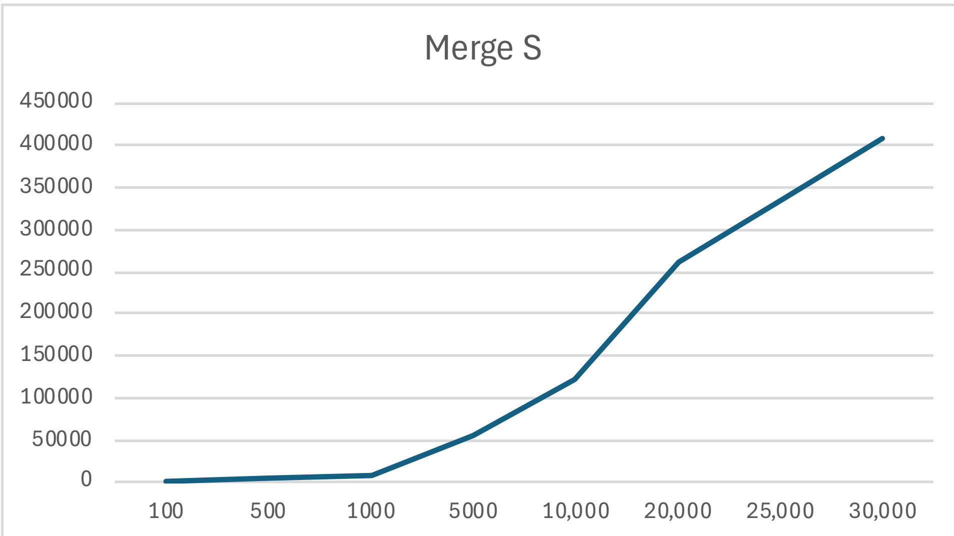
Quick Sort-fp
Theoretical Average Time Complexity: $O(n \log n)$
Analysis: Similar growth to Merge Sort, confirming $n \log n$ performance in average cases.
Slight variations may be due to pivot selection, but trends align with theory.
Conclusion: Results match expected average-case performance of Quick Sort.



Heap Sort
Theoretical Time Complexity: O(n log n)
Analysis: Shows n log n trend, but higher constants than Merge or Quick Sort.
It has complex data structure.
Conclusion: While it follows the expected growth rate.
The overhead leads to slower actual performance, consistent with expectations.



Quick Sort-rp
Theoretical Average Time Complexity: O(n log n)
Analysis: Also exhibits n log n behavior.
Performs similarly or slightly better at larger inputs than Quick S (fp),
suggesting that randomized pivot improves performance slightly by
avoiding worst-case scenarios.
Conclusion: Results align well with theoretical expectations.



Merge Sort
Theoretical Time Complexity: O(n log n)
Analysis: Growing is significantly better than Selection Sort. From 100 to 1000 (10xinput) results in ~16x increase, and from 1000 to 10000 (another 10x) yields ~ 13x increase. This aligns well with nlogn complexity.
Conclusion: Experimental results are consistent with the theoretical complexity.