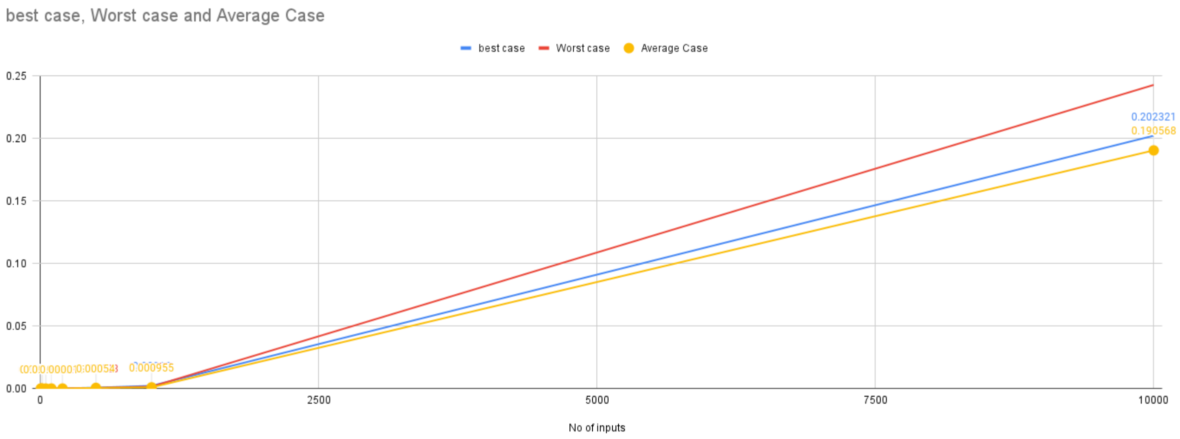
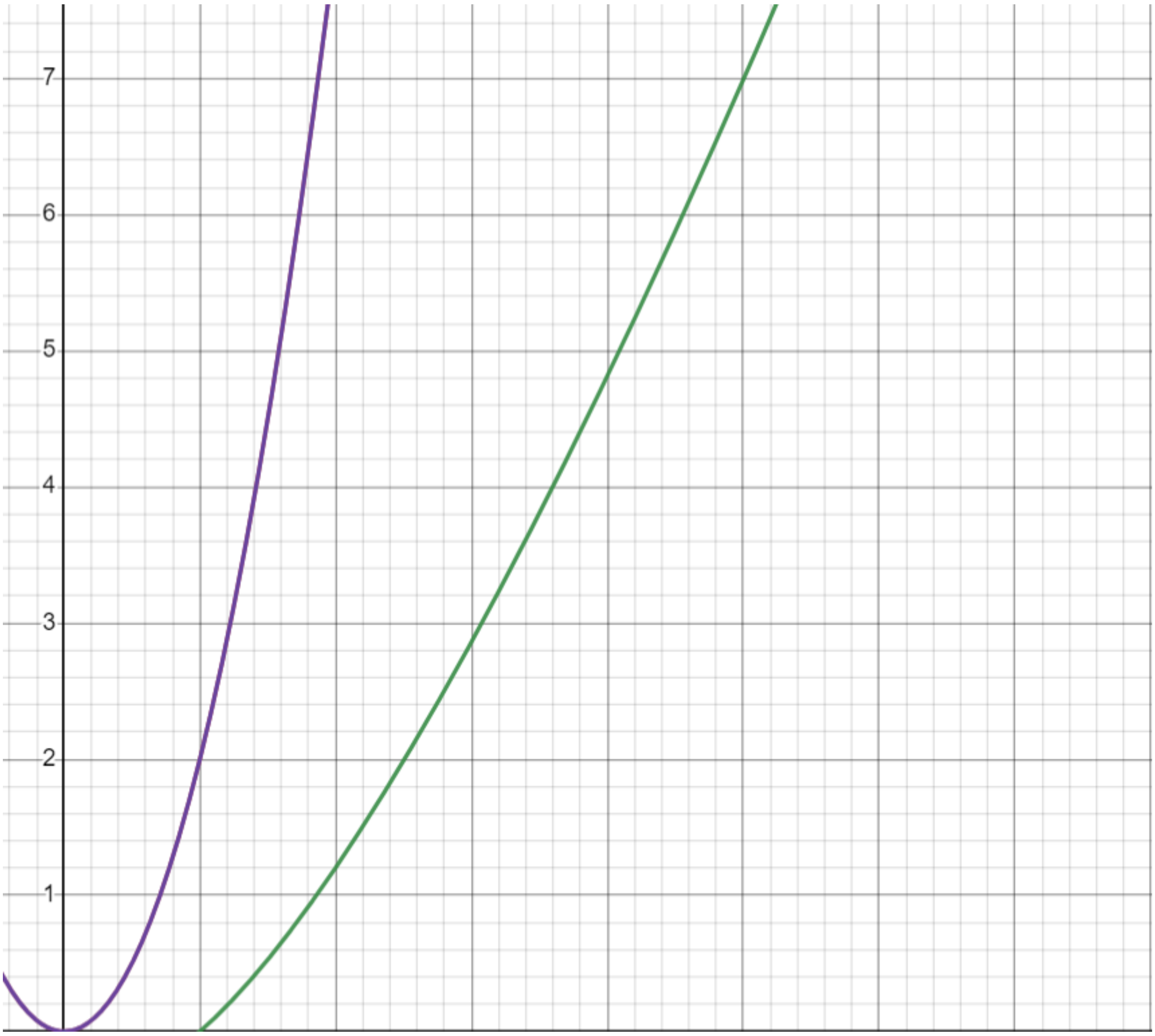
ABHINAV BATTULA **1002210110**



No of inputs	best case	Worst case	Average Case
5	0.000002	0.000002	0.000002
10	0.000003	0.000002	0.000003
20	0.000004	0.000004	0.000003
50	0.000009	0.00001	0.000004
100	0.000027	0.000032	0.00003
200	0.000113	0.000113	0.000015
500	0.000715	0.000523	0.00054
1000	0.00218	0.001553	0.000955
10000	0.202321	0.242965	0.190568

Graph of time complexities



Green represents best case

Blue represents worst and average case

TIME COMPLEXITY FOR RANDOM PIVOT

$T(n)=T(k)+T(n-k-1)+O(n)$

where:

- k is the size of the left subarray after partitioning (elements smaller than the pivot),
- n−k−1 is the size of the right subarray after partitioning (elements greater than the pivot),
- O(n) represents the time complexity of partitioning the array.

Best Case:

pivot always divides the array into subarrays of size (n/2). In this case,

$k=(n/2)$ $T(N)=2T(n/2)+O(n)$

so O(nlogn).

AVG case:

pivot divides the array into subarrays of approximately equal size on average

similar to above so O(nlogn).

Worst case:

In the worst-case scenario, the pivot consistently divides the array into subarrays of size

1 and n−1. This occurs when the pivot is the smallest or largest element in the array

$T(N)=T(0)+T(n-1)+O(n)$

$=O(n^2)$