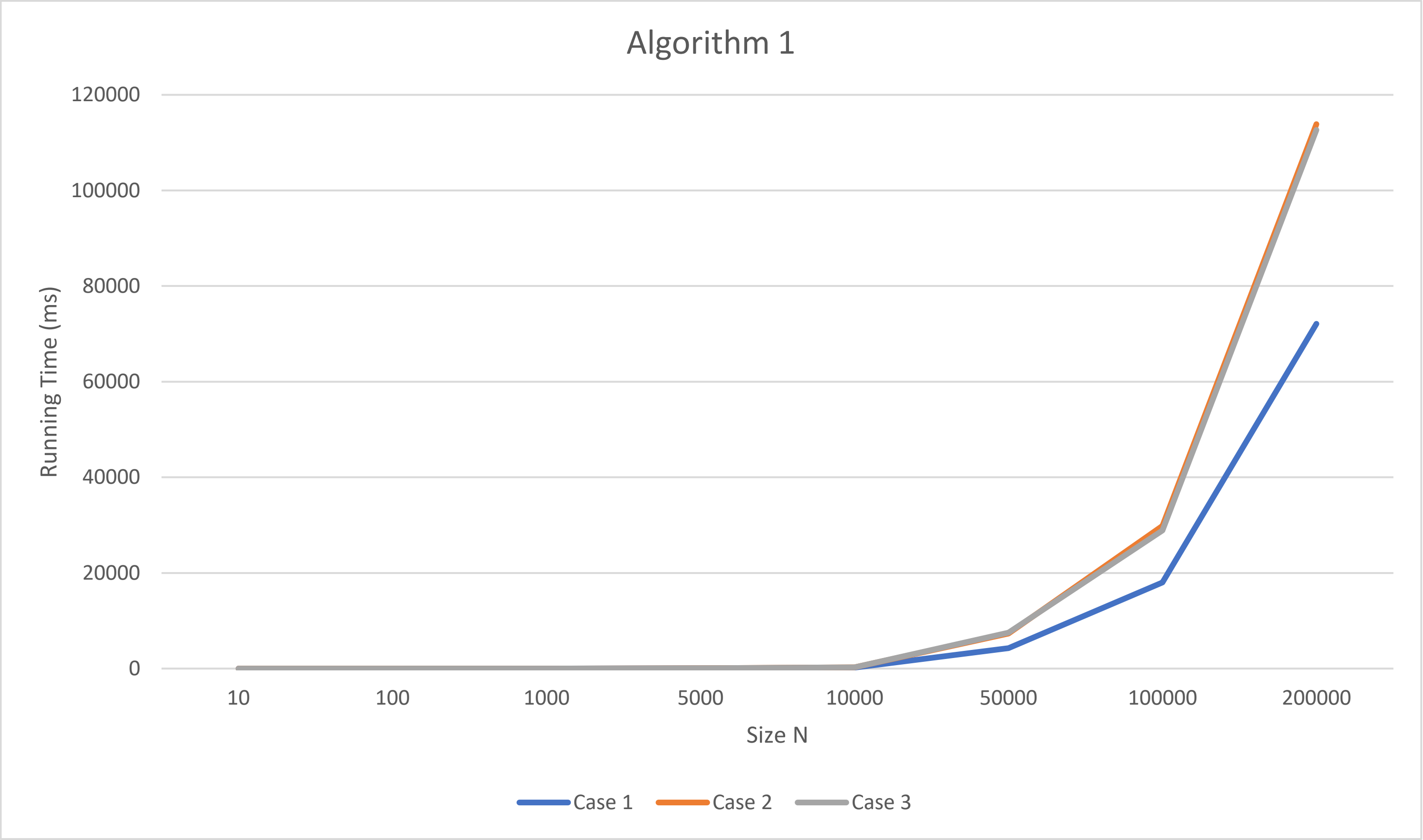
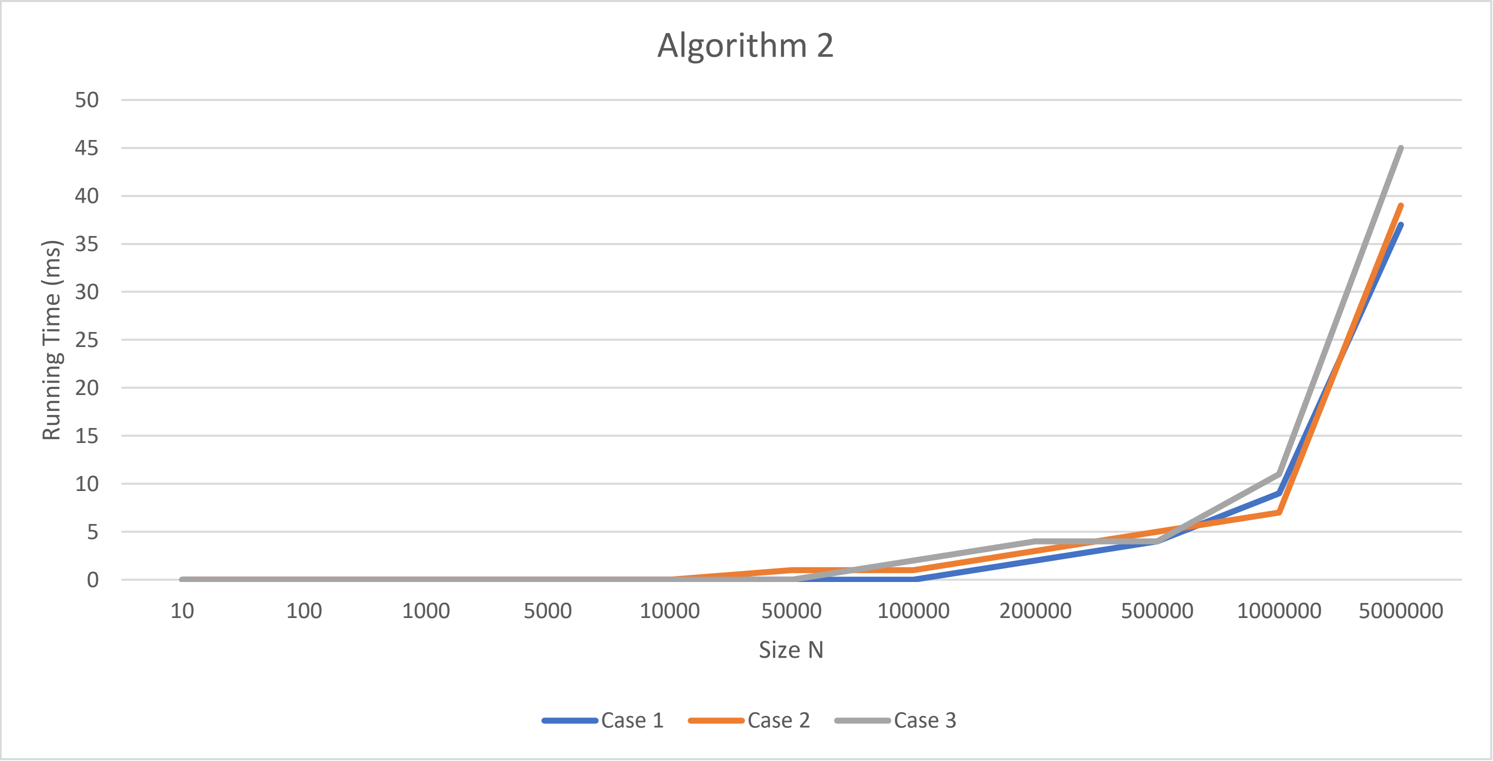
Ata Seren

21901575

CS201 HW2 Report





Specifications of the computer I used to obtain these execution times:

OS: Windows 10 Home

RAM: 16GB

CPU: Intel Core i7-9750H CPU @ 2.60GHz-4.20GHz

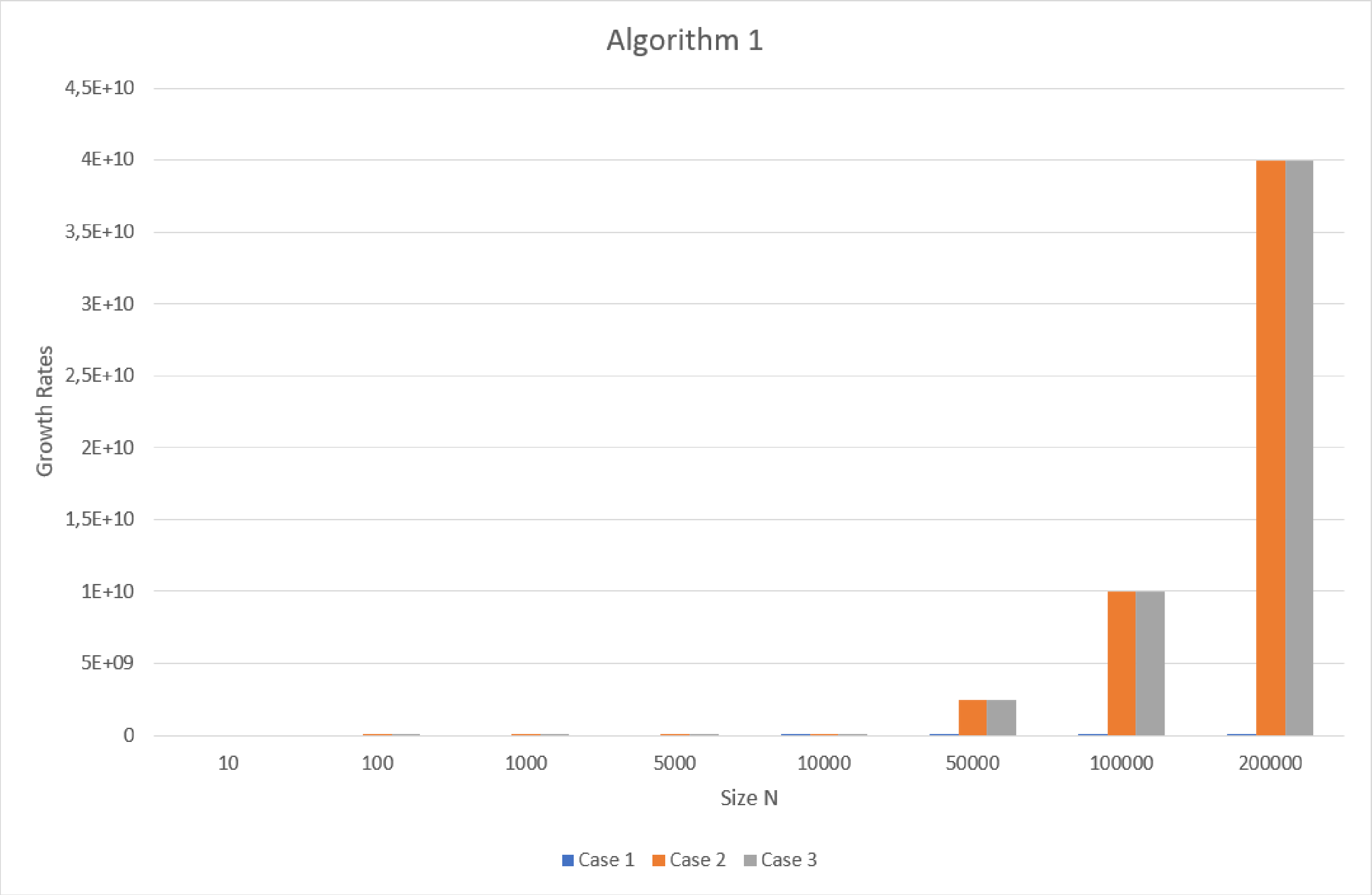
For algorithm 1:

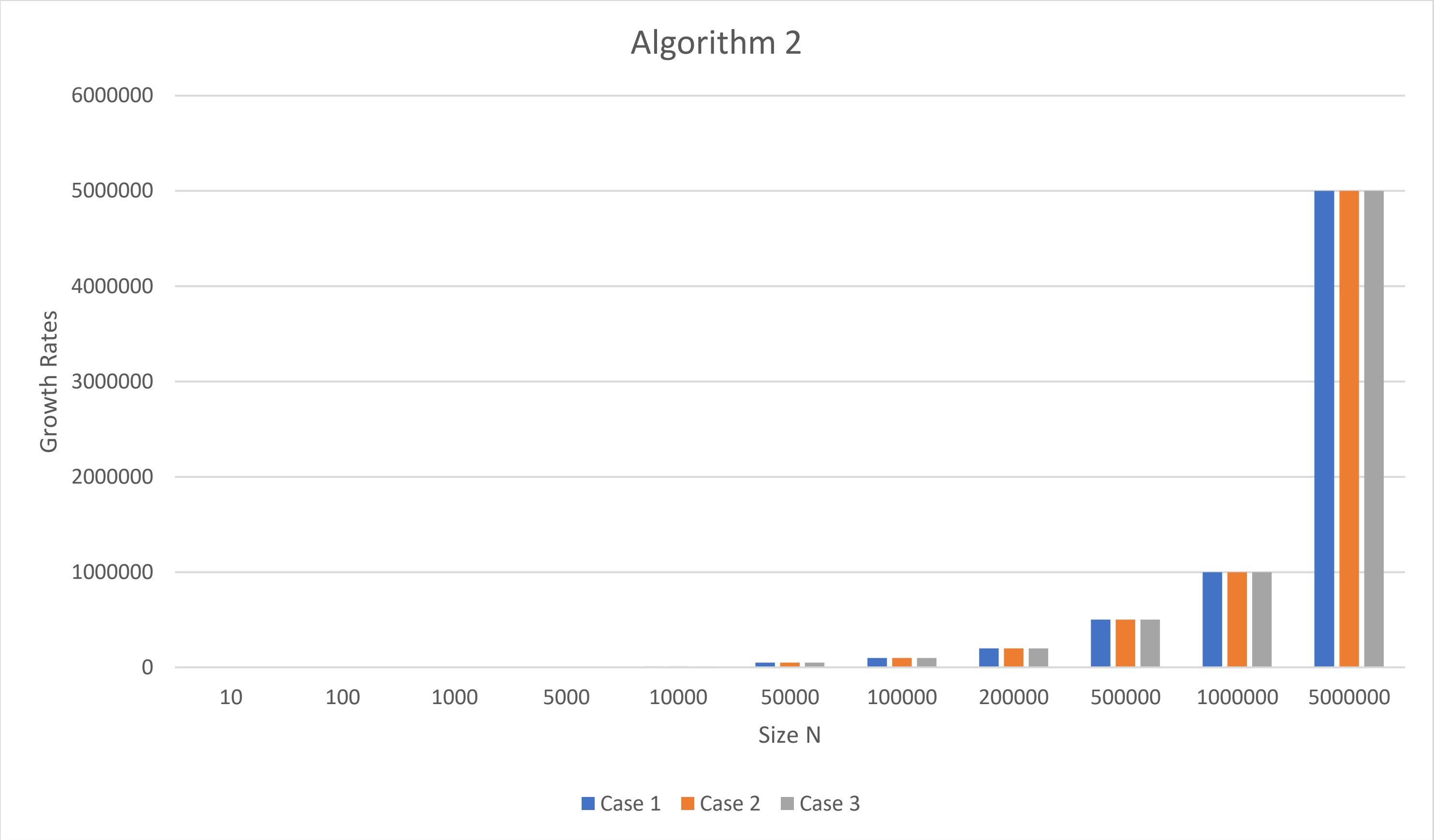
- Best case: 1, Time complexity: O(n)
- Average case: 3, Time complexity: O(n^2)
- Worst case: 2, Time complexity: O(n^2)

For algorithm 2:

- Best case: 1, Time complexity: O(n)
- Average case: 2, Time complexity: O(n)
- Worst case: 3, Time complexity: O(n)

Plot for expected worst cases:





For algorithm 1, according to the plot of running time: best case is 1, average case is 3 and worst case is 2 and according to theoretical analysis: best has O(n) and worst has O(n^2) which confirms the indication of cases. Also, average case is also O(n^2) and in running time plot we can see that case 2 and 3 are close. This can also caused by elements of array in case 3. We don't know how many of the elements are like case 1 or case 2 so, situation of case 3 depends on luck of rand() function I used in my code. For algorithm 2, best case is 1 but we can also show case 2 as best one too. Because they are very close in plot and they have same growth rate of O(n). However, case 3 is worst for sure because of the randomness of elements. Since it doesn't have an order, searching takes more time in case 3, instead of case 1 and 2 which is appending small ones first and appending others without a search mechanism.