|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| All results are in milliseconds. | Algorithm 1 | | | Algorithm 2 | | | Algorithm 3 | | |
|  |  |  |  |  |  |  |  |  |
|  | 50869 | 50300 | 50982 | 82842 | 81728 | 83872 | 69452 | 70746 | 70672 |
|  | 51586 | 51348 | 51821 | 86063 | 84791 | 83935 | 69813 | 72892 | 70729 |
|  | 52818 | 52495 | 52947 | 87104 | 87217 | 86251 | 71705 | 72458 | 72336 |
|  | 54281 | 53878 | 53644 | 88929 | 89085 | 88748 | 72611 | 71972 | 72444 |
|  | 54937 | 55587 | 55182 | 91822 | 91166 | 90557 | 72282 | 73342 | 71516 |
|  | 56685 | 56513 | 56481 | 91866 | 91840 | 93188 | 72926 | 73723 | 72709 |
|  | 57873 | 57449 | 57464 | 93940 | 94237 | 94564 | 70016 | 71058 | 72266 |
|  | 57939 | 57860 | 58372 | 94525 | 94674 | 95833 | 70771 | 71002 | 71349 |
|  | 59251 | 59582 | 59541 | 97760 | 96843 | 97470 | 70543 | 70857 | 73188 |
|  | 60753 | 61752 | 60476 | 99126 | 100193 | 98633 | 73201 | 72956 | 71324 |

**CS201 Homework-2 Report**

**Oğuz Akın**

**22002893**

**Section-1**

Results of all the test runs are represented in the following chart:

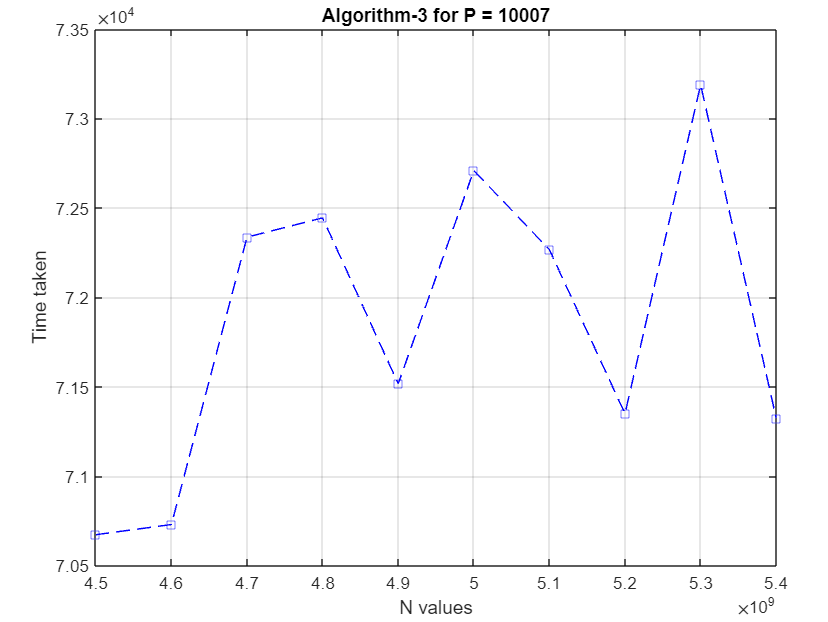
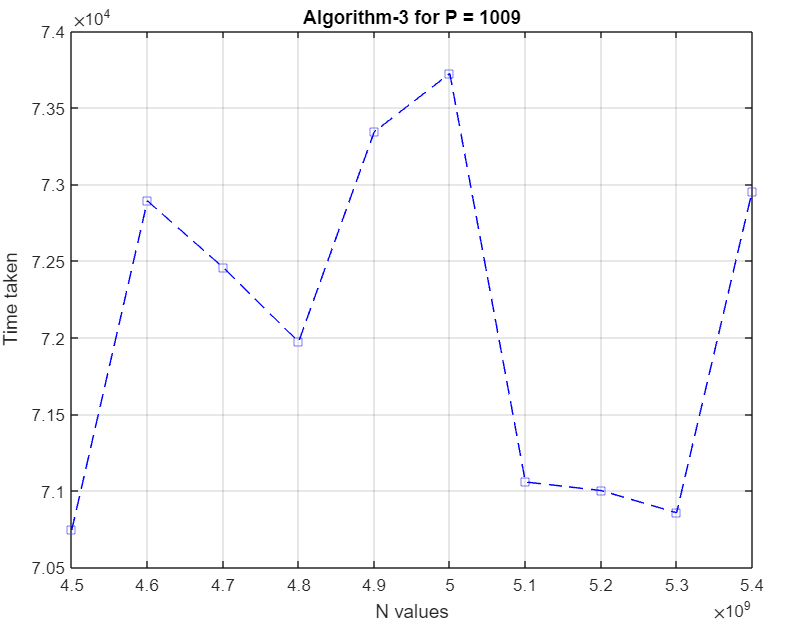
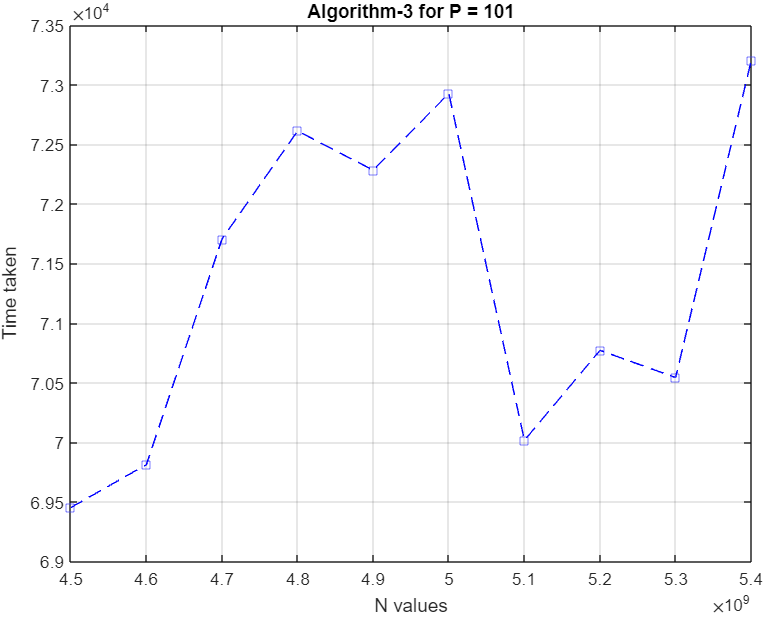
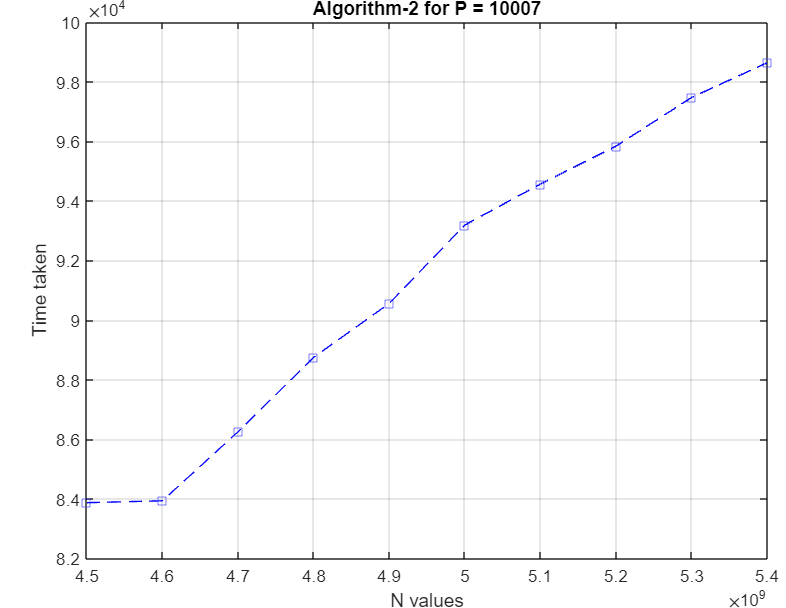
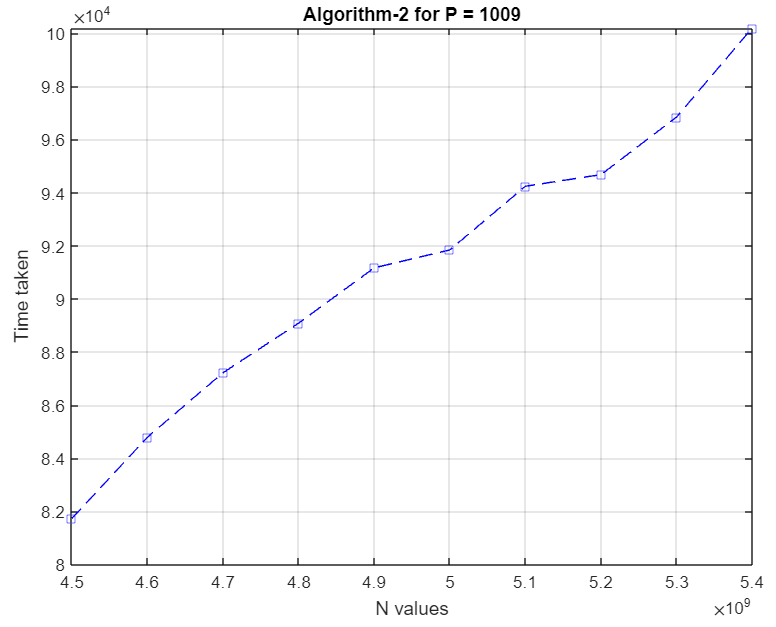
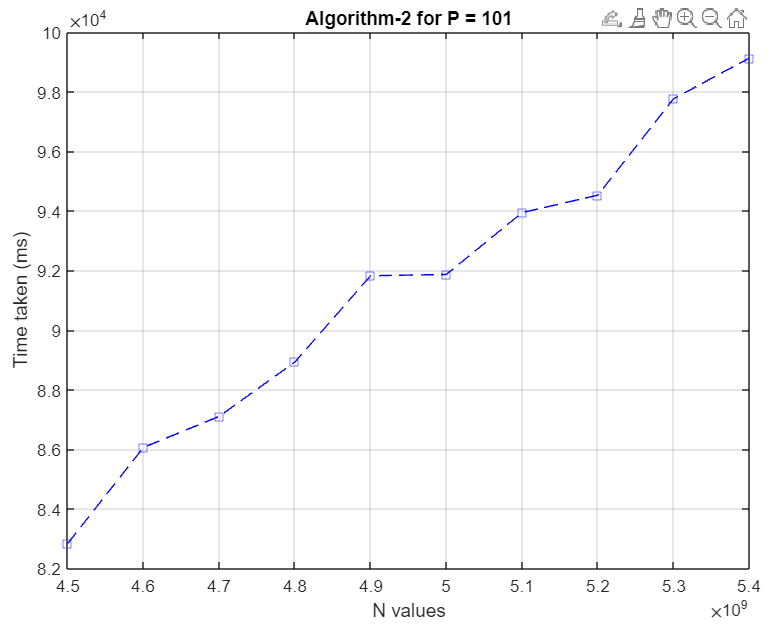
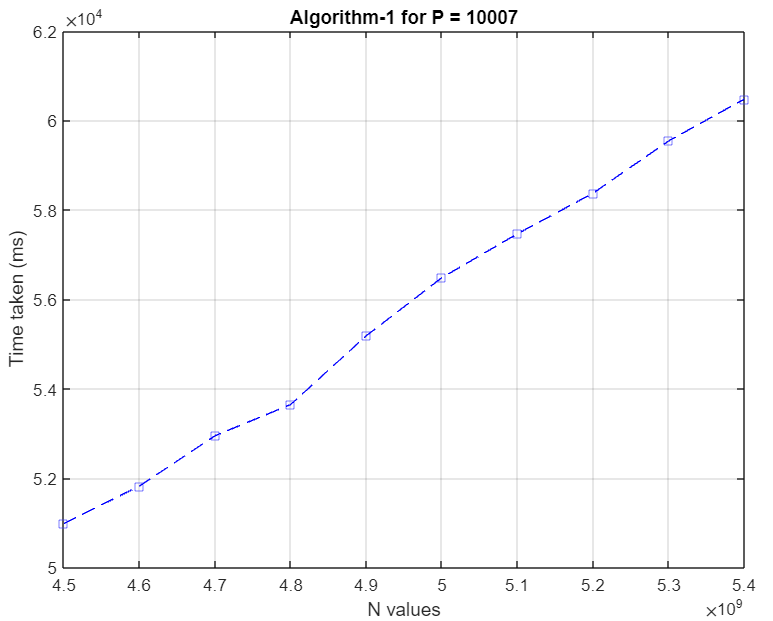
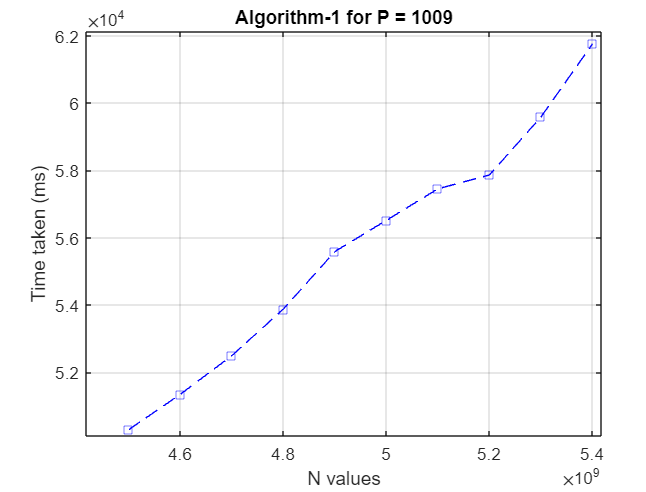
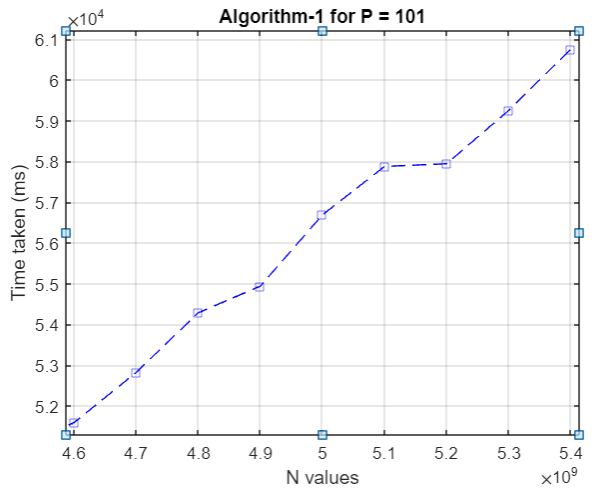
These results were obtained with the computer that has the following specifications:

RAM: 16 GB. (Available: 15.9 GB)

Processor: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz.

System Type: 64 bit operating system, x64 based processor.

**GRAPHS (Time taken is always measured in milliseconds)**



**DISCUSSION**

Time complexity for the first algorithm is simply calculated by considering the for loop only as the other statements take up O(1) time while the loop takes O(N) time. Time complexity of the second algorithm is similar to the first one but with a slight difference: if there were to be an equation for , it would stop looping and give us the result and thus take less time. However, since we are only considering worst case time complexities, we will imagine that the loop iterates all the way to the end. So, the second algorithm will have time complexity O(N). Time complexity for the third algorithm is different than both its predecessors as this algorithm has a recursive call. Even if the n value were odd or even, the function is called twice to make up the original number and the problem size is always divided into two when passed as a parameter. So, this function has the time complexity O() which is equal to O().