Athul Shibu Kurian
1002235448

CSE 5311-005 DESIGN & ANALYSIS OF ALGORITHMS
Hands On 3

1. $T(n) = c_1(1) + c_2 \sum_{i=1}^{n+1}(1) + c_3 \sum_{i=1}^{n} \sum_{j=1}^{n+1} 1 + \ c_4 \sum_{i=1}^{n} \sum_{j=1}^{n}(1)$

   $= c_1 + c_2(n+1) + c_3(n)(n+1) + c_4(n^2)$

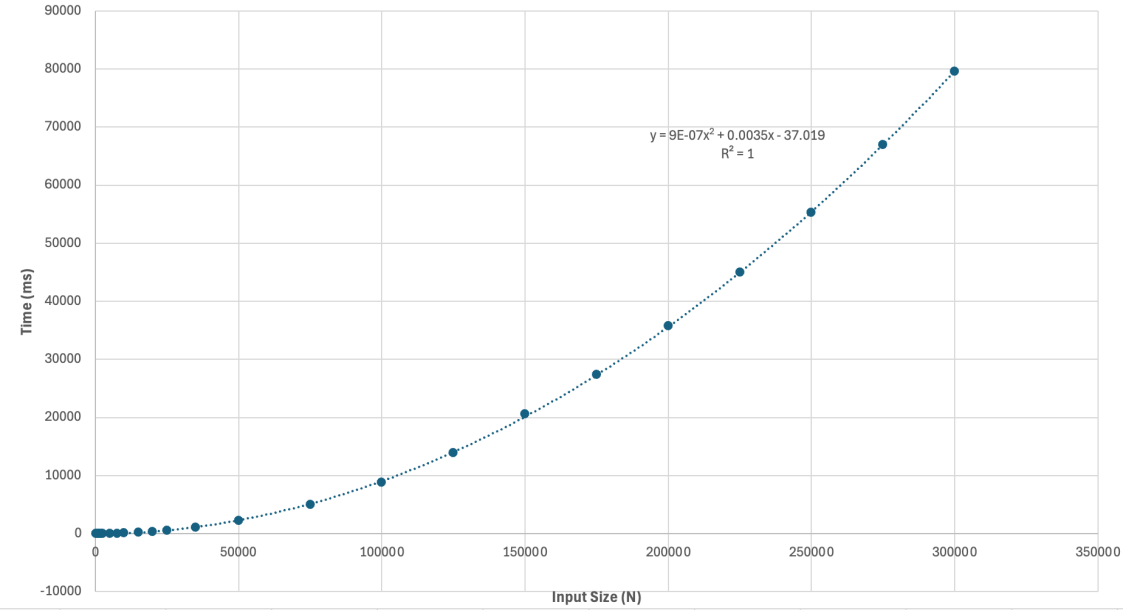
   $= c_1 + c_2(n+1) + c_3(n^2+n) + c_4(n^2)$

   $= (c_3 + c_4)n^2 + (c_2 + c_3)n + (c_1 + c_2)$

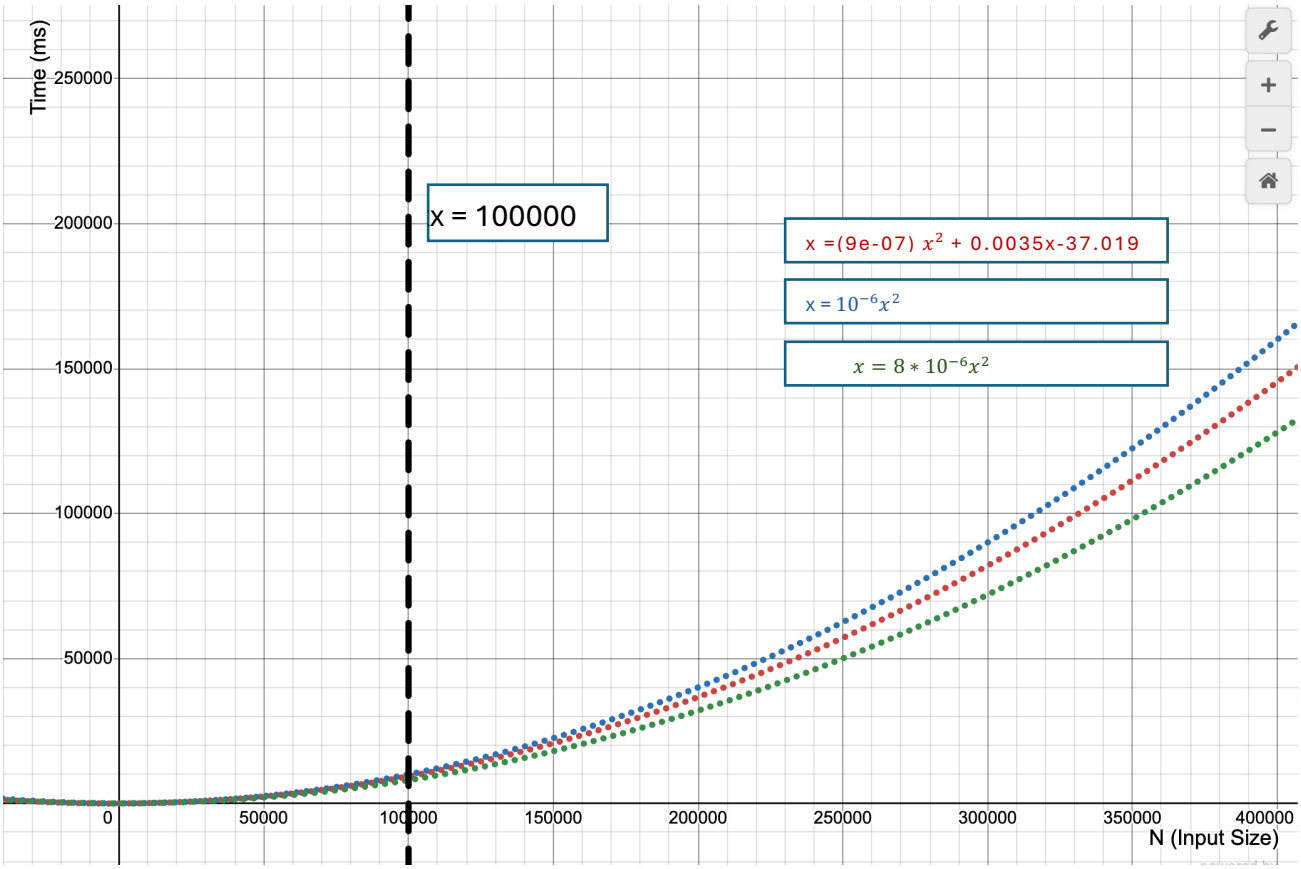
   $= \Theta(n^2)$

2.

Input Size (N)	Runtime (ms)
250	1
500	1
1000	4
1250	6
1500	7
2000	9
2500	17
5000	37
7500	68
10000	92
15000	196
20000	346
25000	549
35000	1065
50000	2210
75000	4975
100000	8848
125000	13917
150000	20581
175000	27379
200000	35794
225000	44974
250000	55306
275000	66963
300000	79645



3.



From the above graph, a good upper bound for f(x) = $(9 * 10^{-7})\ x^2 + 0.0035x - 37.019$ would be f(x) = $10^{-6}x^2$ and a good lower bound would be $f(x) = (8 * 10^{-6})x^2$.

From this we can say, $O(f(x)) = x^2, \Omega(f(x)) = x^2, and\ \Theta(f(x)) = x^2$

4.

From the above graph, a good value for n_0 would be $n_0 = 100000$ since the curve $f(x)$ is sandwiched between the upper and lower bounds for all values of $x \geq n_0$

4. Will this increate how long it takes the algorithm to run (e.x. you are timing the function like in #2)?
   The addition of the lines y = 1, and y = i + j; will increase the time slightly. Escpecially the second line, y = i + j is inside of a loop, so it has to be executed n more times. However, the time will not increase drastically since the asymptotic complexities will remain the same.

5. The addition of the lines y = 1, and y = i + j will add a constant term and a linear term respectively to the runtime function T(n) found in 1. However, since the dominant term n^2 is unchanged, the asymptotic complexity will remain the same.