NEW MEXICO TECH
COMPUTER SCIENCE AND ENGINEERING

CSE/IT 122: Blackbox
Summary

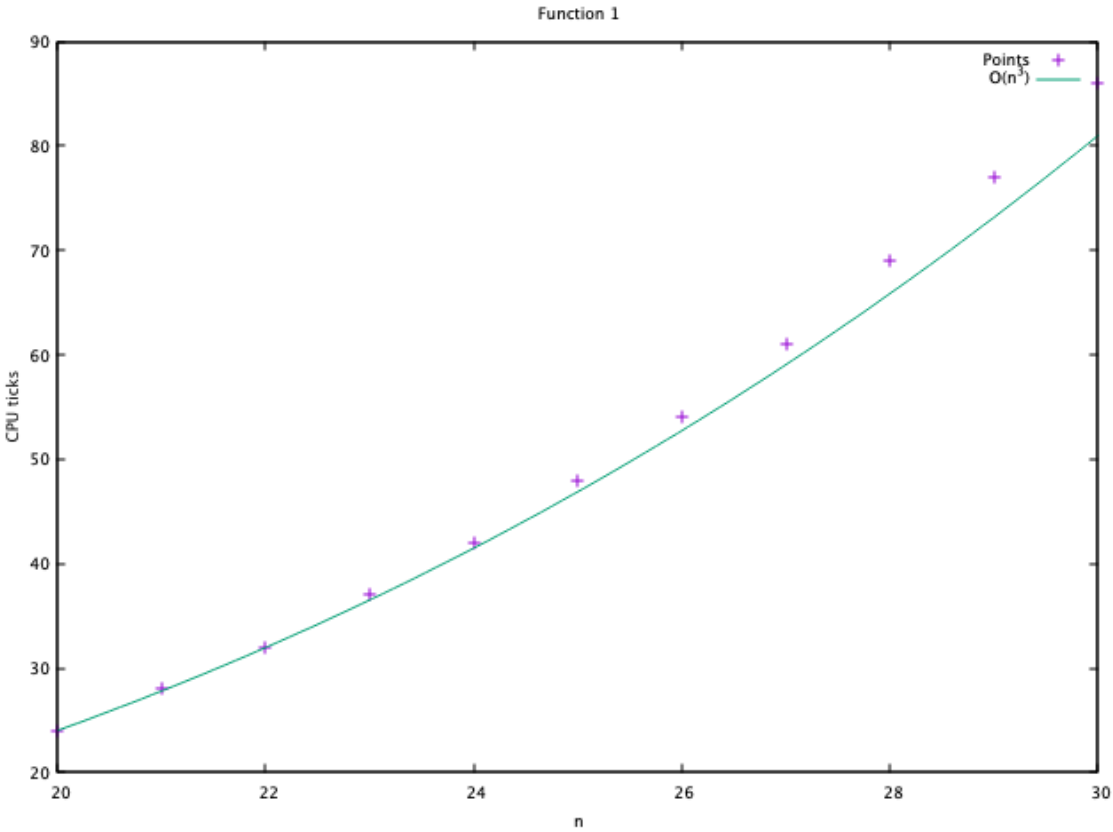
Michael Mingyang Fan

July 2, 2019

*Note: It may be difficult to see the data points. For all cases, at least 5 values were tested 5-10 + times to find the most consistent and average runtimes of the functions.

1 FUNCTION 1

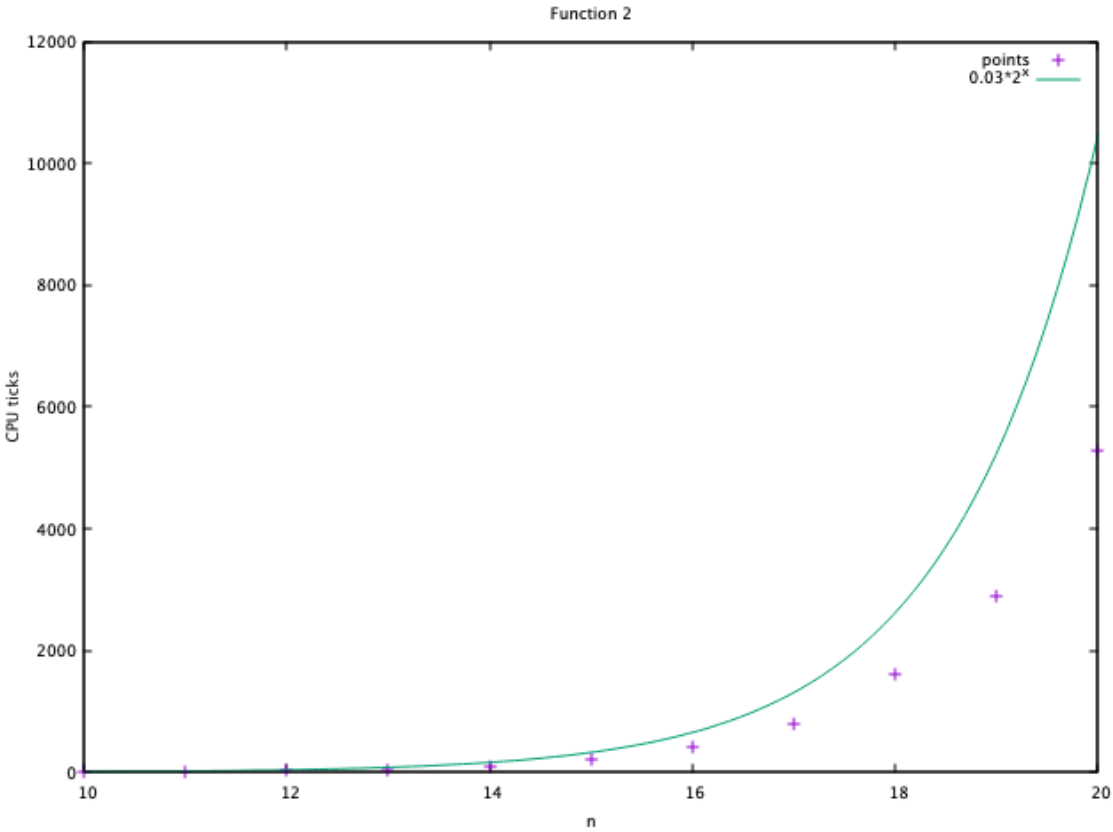
Runtime: $O(n^3)$



Function 1 was tested multiple times with values ranging from n = 20 to n = 30 (all values tested 5 times each for consistency). These were averaged and the plot above was made. Fitting it with gnuplot, it was found that the function $c * n^3$ fit to an asymptotic error of 0.07554% when $c = 0.003$.

2 FUNCTION 2

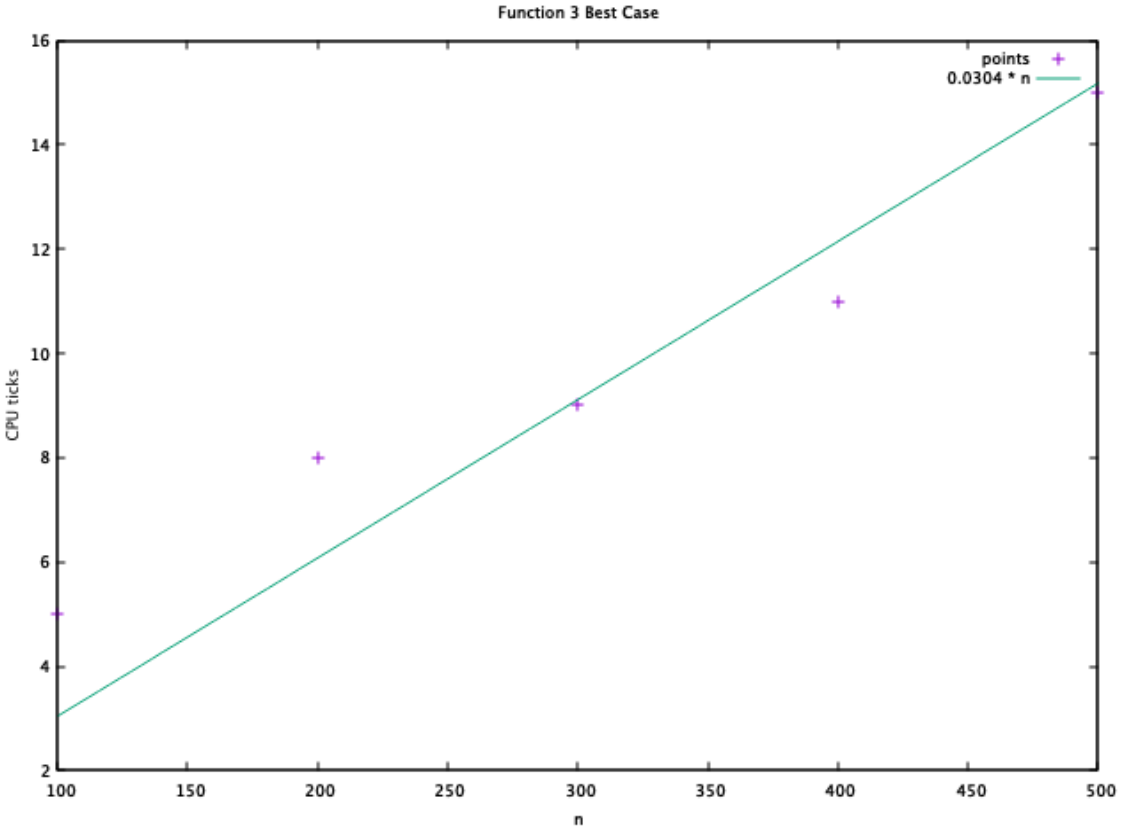
Runtime: $O(2^n)$

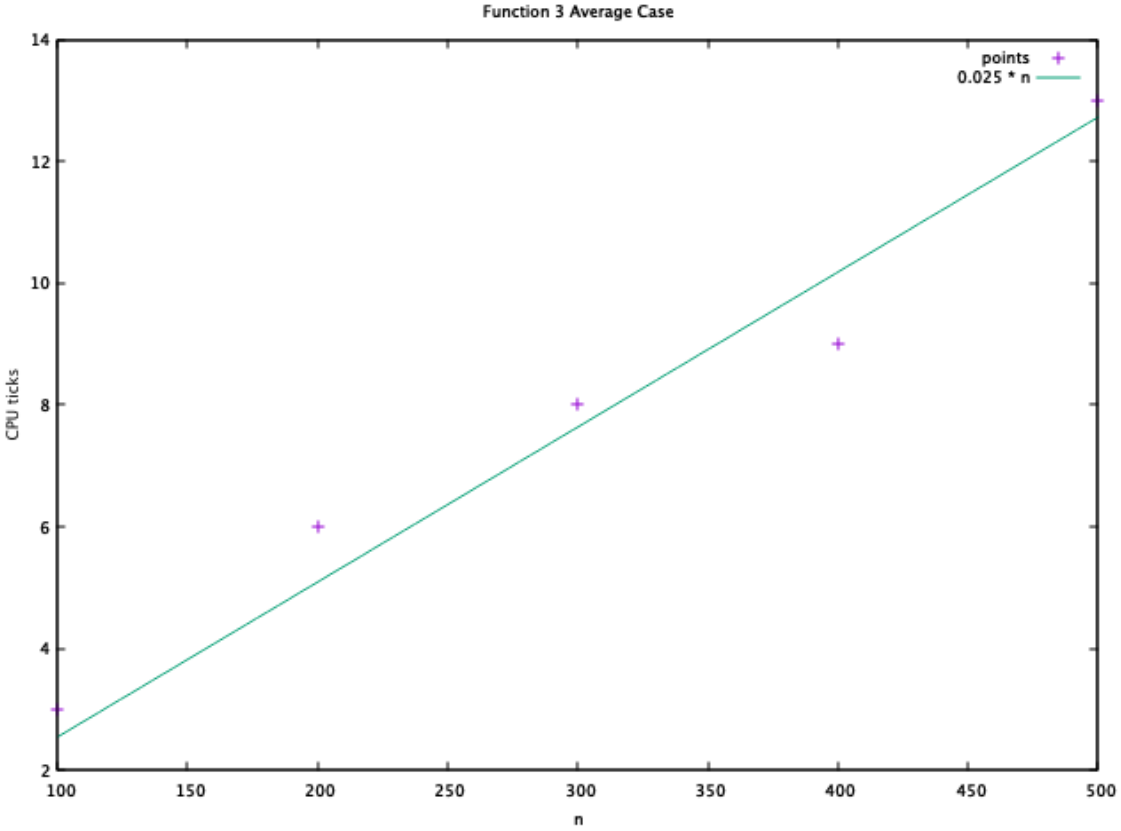


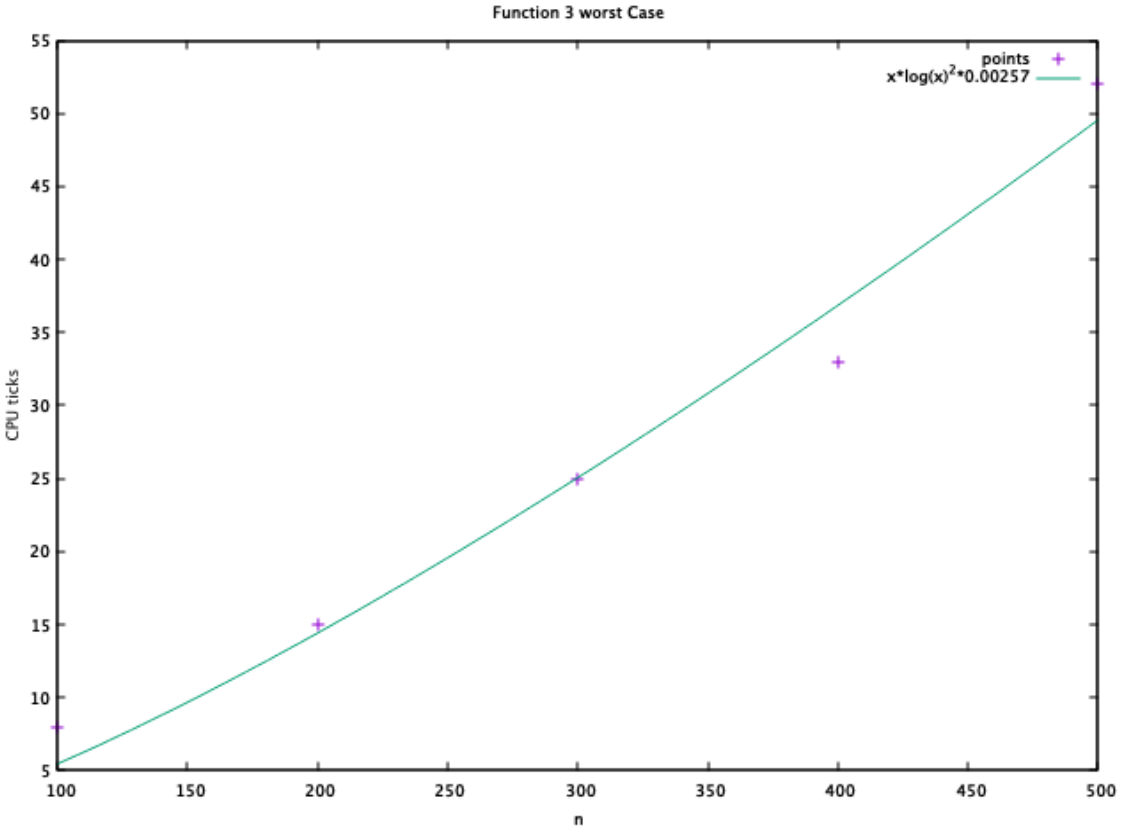
Function 2 was tested with values of n = 10 to n = 20. The runtime was consistent with the $c * 2^n$ fit, where $c = 0.01$ with an asymptotic error of 0.505%.

3 FUNCTION 3

Best Case Runtime: $O(n)$
Average Case Runtime: $O(n)$
Worst Case Runtime: $O(nlog(n)^2)$


Function 3 Best Case
points
0.0304 * n
CPU ticks
n
16
14
12
10
8
6
4
2
100
150
200
250
300
350
400
450
500

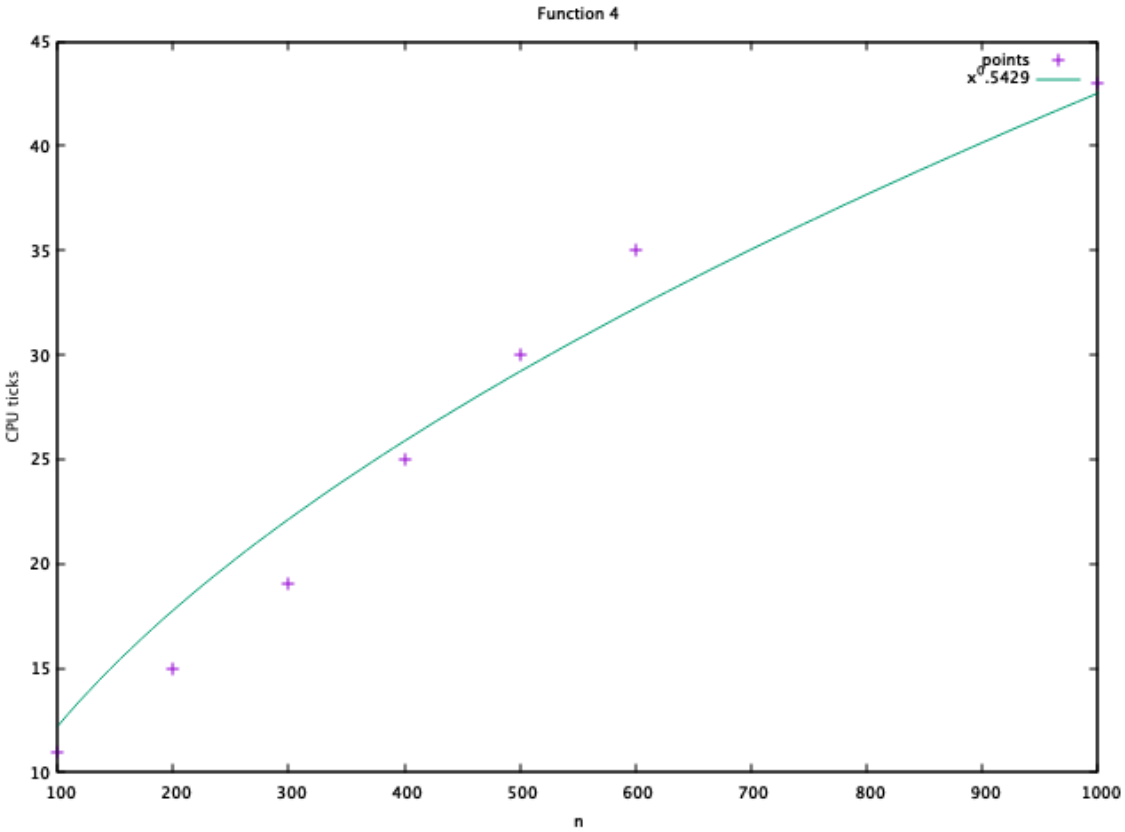

Function 3 Average Case
points
0.025 * n
CPU ticks
n
14
12
10
8
6
4
2
100
150
200
250
300
350
400
450
500



Function 3 was a sorting algorithm, so it was tested for values of n = 100 to n = 500 for 3 sorting cases: in order list (best case), reverse order list (worst case), and random list (average case). The best case was consistent with $O(c*n)$, where $c = 0.0304$ with an asymptotic error of 6.634%. The average case was consistent with $O(c*n)$, where $c = 0.025$ with an asymptotic error of 4.301%. The worst case is $O(nlog(n)^2)+c$, with $c = 0.000257$ with an asymptotic error of 3.846%.

4 FUNCTION 4

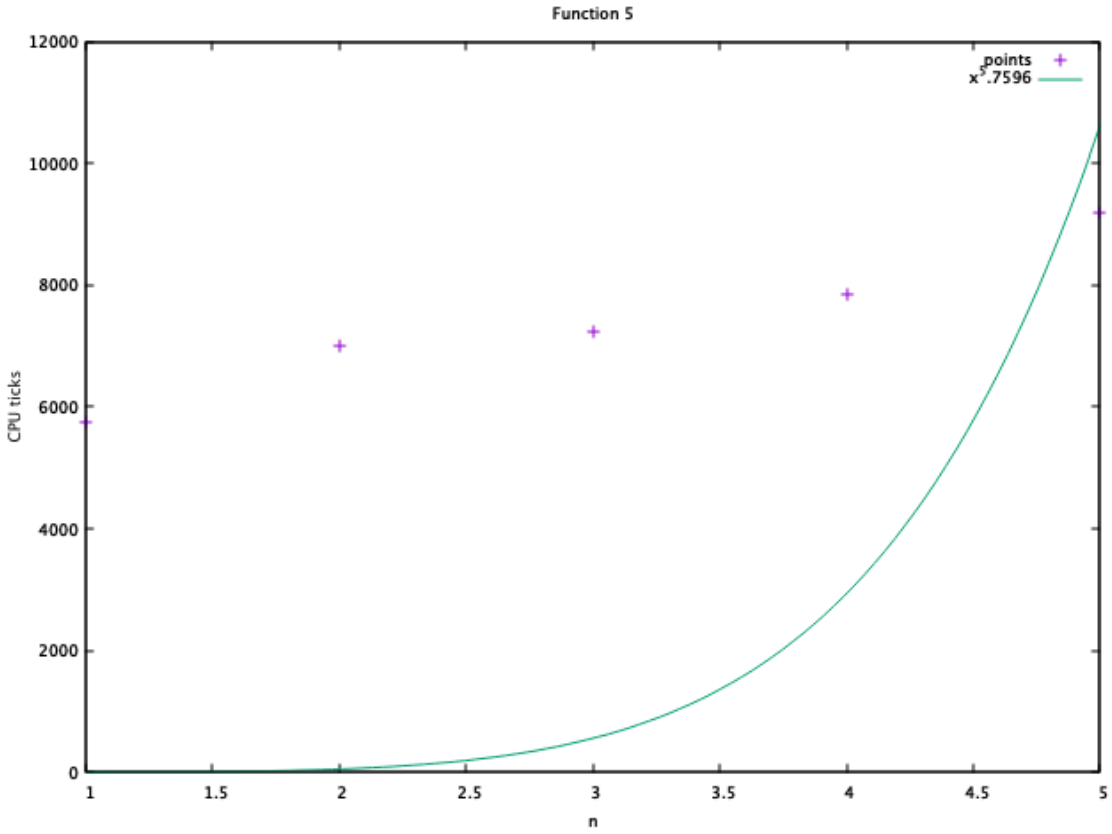
Runtime: $O(x^c)$



Function 4 was tested with values from n = 100 to n = 1000. This algorithm was consistent with the n^c fit, where $c = 0.5429$ with an asymptotic error of 0.859%.

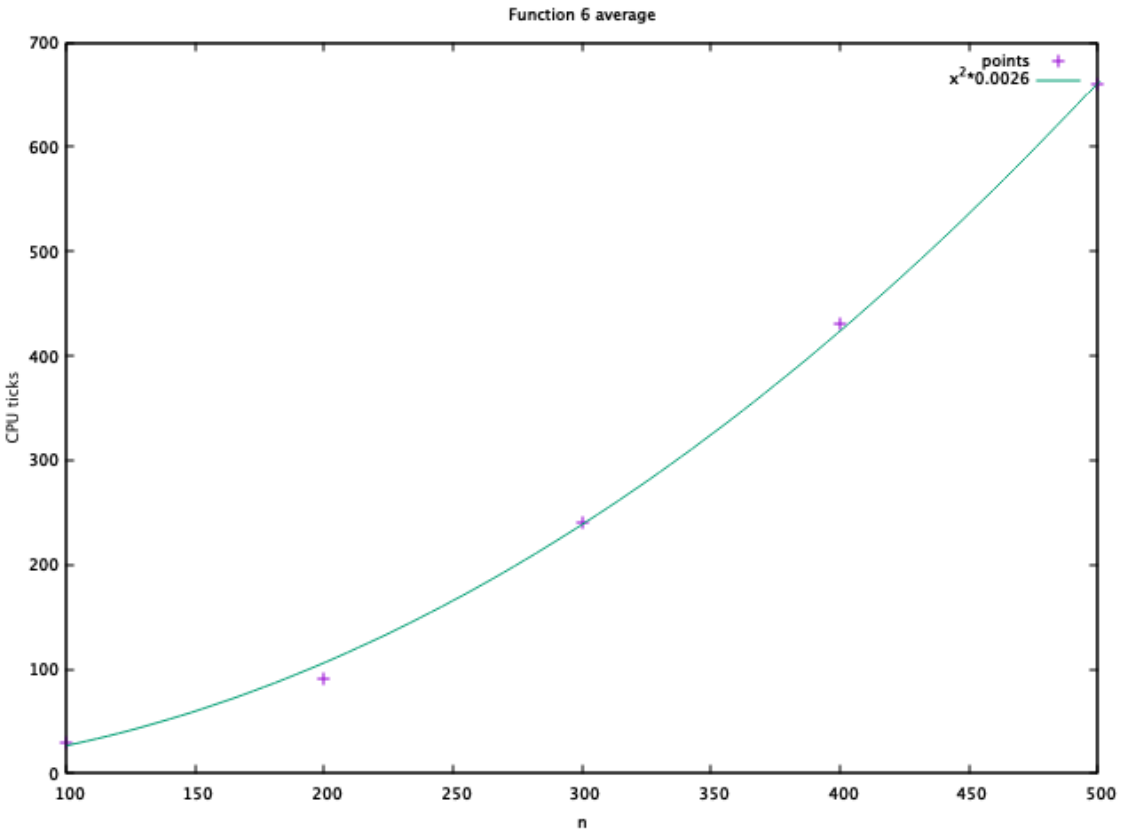
5 FUNCTION 5

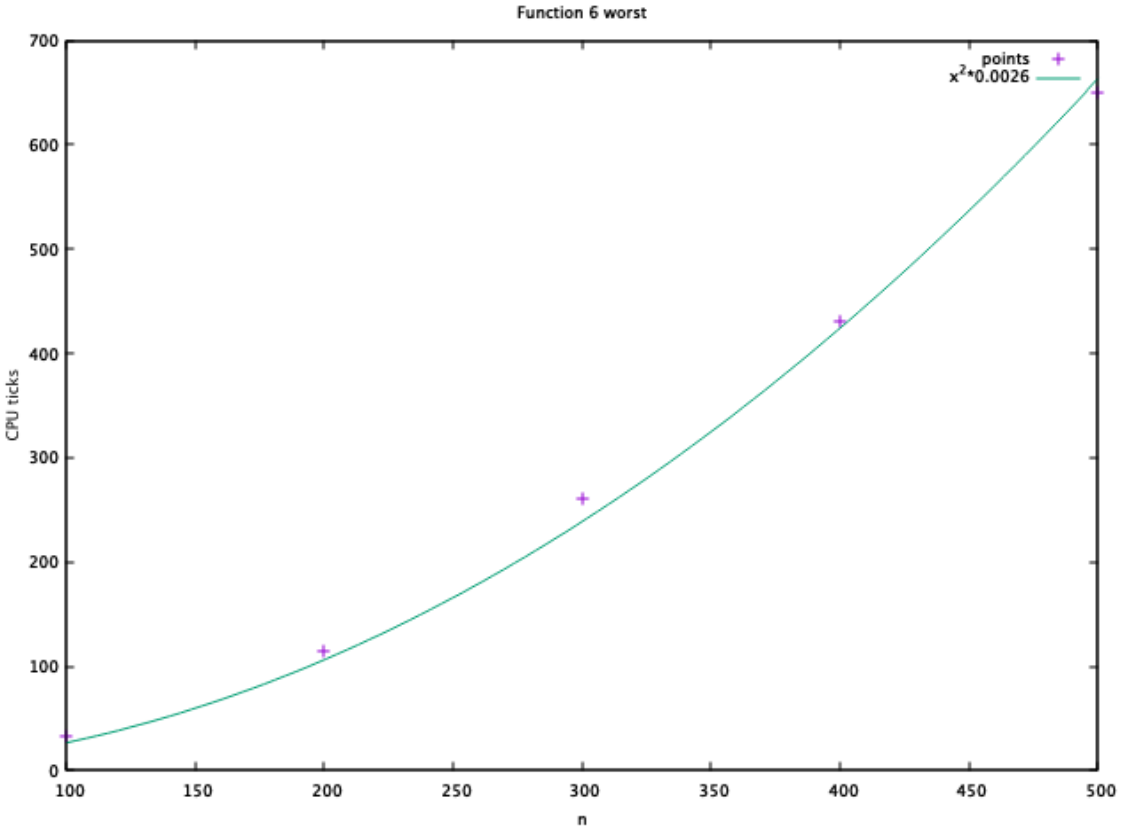
Runtime: $O(n^c)$



Function 5 was tested with values from n = 1 to n = 5. This algorithm was consistent with the n^c fit, where $c = 5.7596$ with an asymptotic error of 6.071%.

6 FUNCTION 6

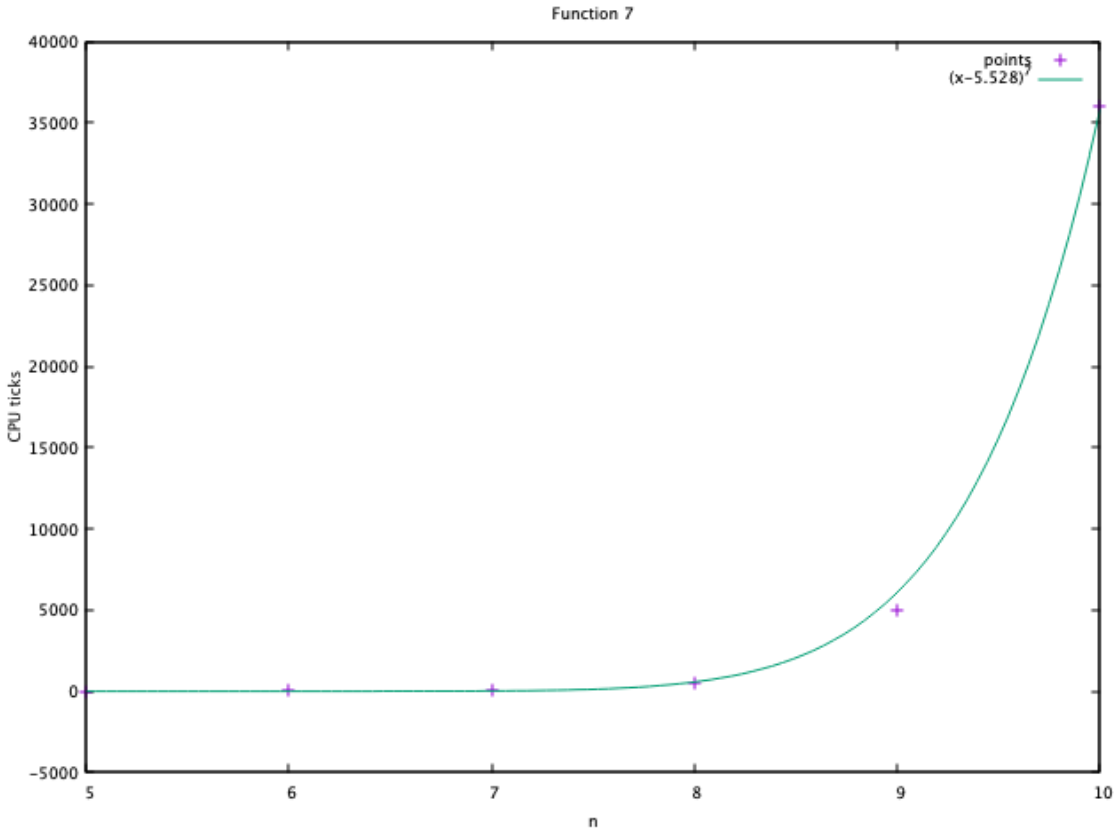

Function 6 average
points
x^2*0.0026
CPU ticks
n
0
100
200
300
400
500
600
700
100
150
200
250
300
350
400
450
500


Function 6 worst
points
x^2*0.0026
CPU ticks
n
0
100
200
300
400
500
600
700
100
150
200
250
300
350
400
450
500

Best Case Runtime: $O(1)$
Average Case Runtime: $O(n^2)$
Worst Case Runtime: $O(n^2)$
Function 6 was a sorting algorithm, so it was tested for values of n = 100 to n = 500 for 3 sorting cases: in order list (best case), reverse order list (worst case), and random list (average case). The best case was consistent with $O(1)$, where the runtime was mostly a constant 1-2 ticks. The average case was consistent with $O(c * n^2)$, where $c = 0.0026$ with an asymptotic error of 1.056%. The worst case is $O(n^2)$, with $c = 0.0026$ with an asymptotic error of 1.662%.

7 Function 7

Runtime: $O(x^c) or O(x^7)$



Function 7 was tested with values of n = 5 to n = 10. The values of this algorithm grew very quickly, so small values were tested. This algorithm is consistent with $(x-c)^7$, where $c = 5.528$ with an asymptotic error of 0.1573%.