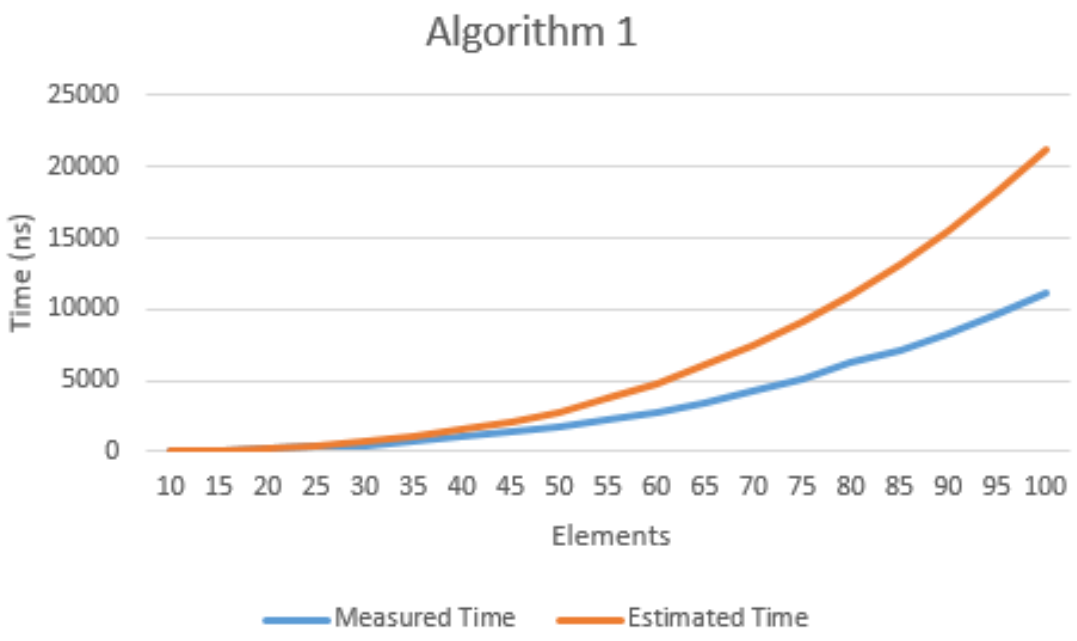
Input Data

```
algorithm-1,algorithm-2,algorithm-3,algorithm-4,T1(n),T2(n),T3(n), T4(n)
300,300,400,100,344,127,22,19
1000,600,600,200,980,273,32,29
1700,1000,1100,100,2126,474,44,38
3500,1600,1000,300,3932,730,55,48
4400,2100,1400,200,6548,1041,67,57
7200,3800,1900,400,10124,1407,80,67
10100,3800,1900,300,14810,1828,92,76
13400,4700,2100,400,20756,2304,105,86
17500,5700,2400,500,28112,2835,117,95
21900,6800,2700,500,37028,3421,130,105
28000,8100,3000,500,47654,4062,143,114
34400,9600,3100,500,60140,4758,156,124
41900,11200,3400,600,74636,5509,170,133
50500,12900,3800,600,91292,6315,183,143
62300,14400,4100,800,110258,7176,197,152
70900,16200,4400,800,131684,8092,210,162
82900,18200,4500,800,155720,9063,224,171
97000,20300,5000,900,182516,10089,238,181
111400,22600,5100,900,212222,11170,251,190
```

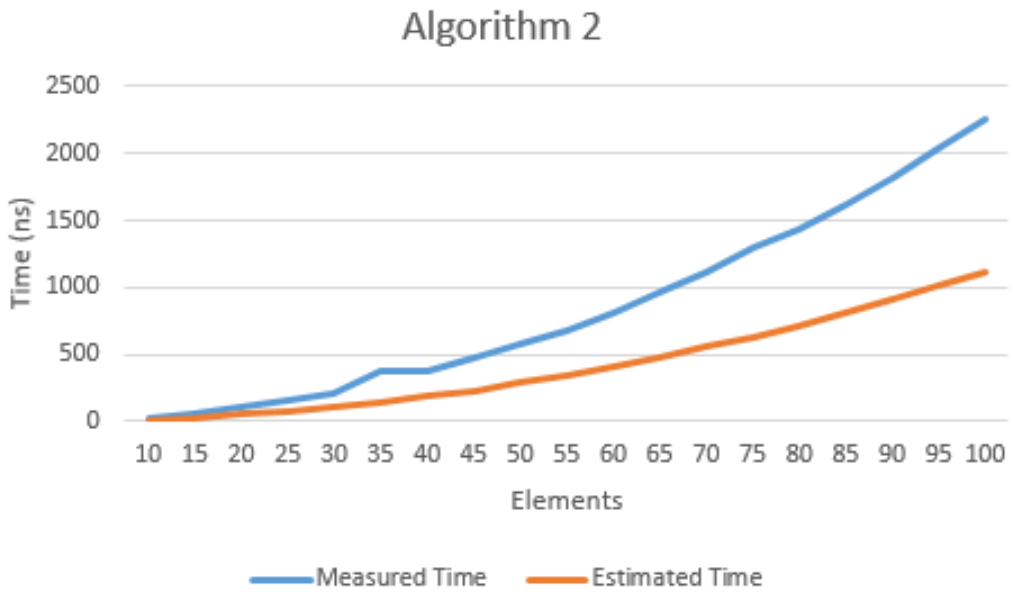
Note: All inputs were scaled by a factor of 10^{-1} for the purpose of charting. All charts exhibit consistent growth rate, and any differences in scale are likely due to the lack of cost attributed to looping mechanisms and certain function calls. The measured time was calculated by saving a clock time before the execution of the algorithm, and after the execution subtracting the current clock value from the saved clock value. An example of the two lines of code are as follows:

```
clock_t timet1 = clock()
// Execute algorithm
double elapsed_timet1 = double(clock() - timet1) / CLOCKS_PER_SEC;
```

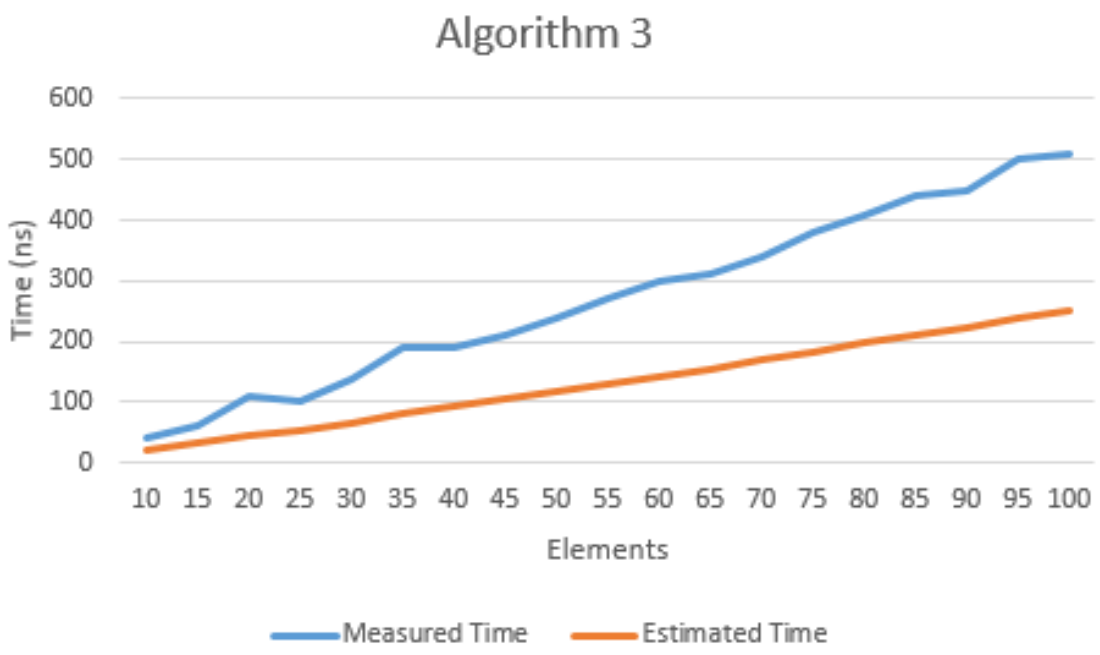
All elapsed times were then scaled by a factor of 10^5. The estimated times were calculated using the corresponding $T(n)$.



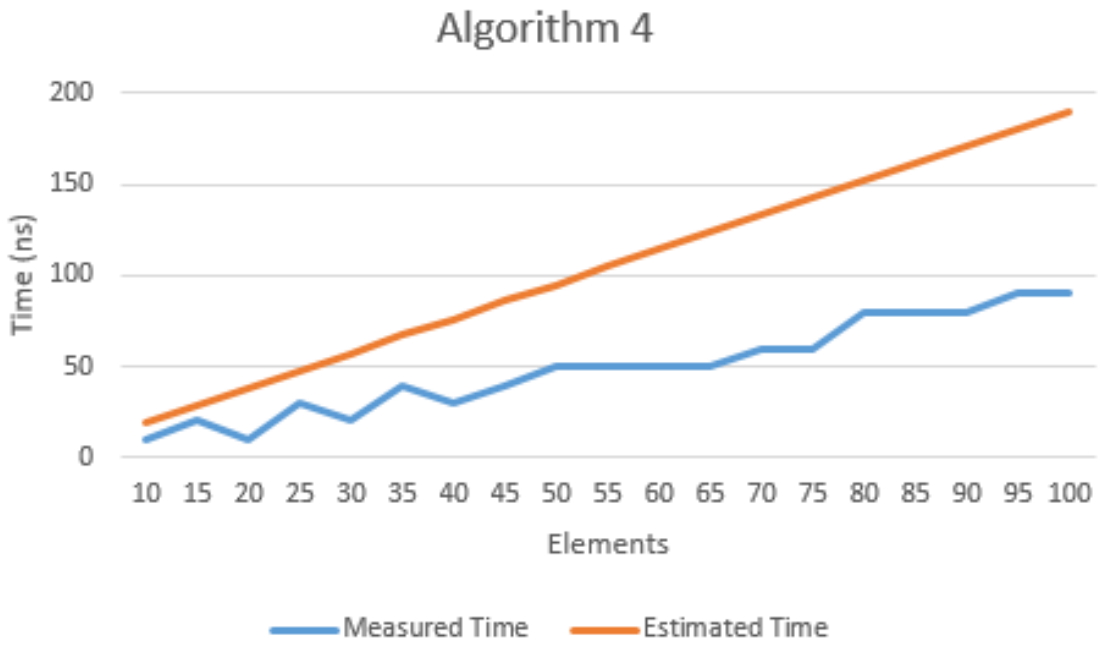
Explanation of Graph: As the number of elements inserted to n grows, the estimated time grows larger than the measured time. This difference in scale is likely due to overestimating the cost for the max function. The growth rate is similar, and charted onto an input with more numbers and in a scale of seconds would not reveal a large difference. T(n) used for estimated time: $2n^3 + 12n^2 + 22n + 24$ where n is the number of elements.



Explanation of Graph: As the number of elements inserted to n grows, the measured time grows larger larger than the estimated time. This is likely due to a lack of cost for the looping mechanisms. The growth rate is similar, and charted onto an input with more numbers and in a scale of seconds would not reveal a large difference. T(n) used for estimated time: $11n^2 + 17n + 8$ where n is the number of elements.



Explanation of Graph: As the number of elements inserted to n grows, the measured time grows larger larger than the estimated time. This is likely due to a lack of cost for the looping mechanisms and miscalculation of the cost for max(a,b,c). The growth rate is similar, and charted onto an input with more numbers and in a scale of seconds would not reveal a large difference. T(n) used for estimated time: $2(T(n/2)) + 23n/2 + 31 + T(1)$ where n is the number of elements.



Explanation of Graph: As the number of elements inserted to n grows, the estimated time grows larger larger than the measured time. This is likely due to the amount of time being so small that single nanosecond differences in time measurement would appear as large differences. The growth rate is similar, and charted onto an input with more numbers and in a scale of seconds would not reveal a large difference. T(n) used for estimated time: $16n + 5$ where n is the number of elements.