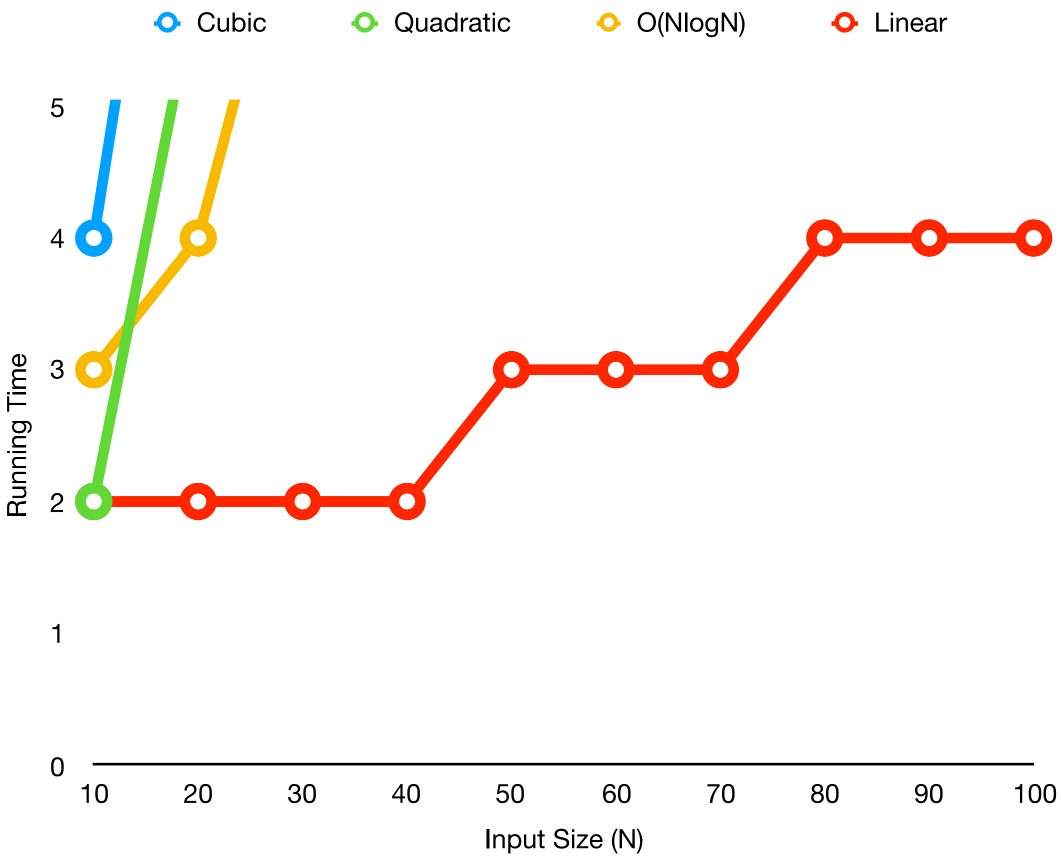
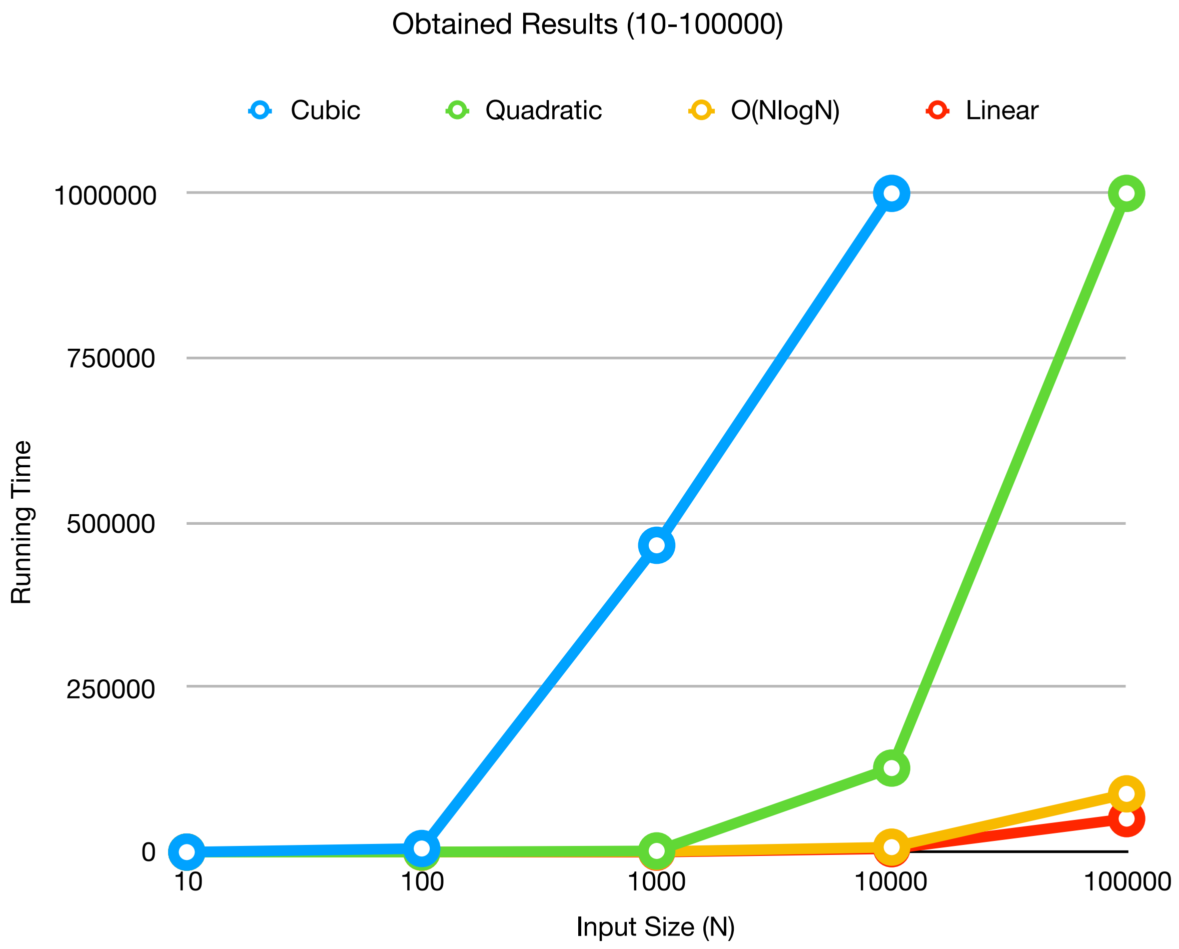
```
int main() {

    int size1 = 10, size2 = 100, size3 = 1000, size4=100000, size5 = 1000000 ;
    int *array1 = new int[size1], *array2 = new int[size2], *array3 = new int[size3], *array4 = new
int[size4], *array5 = new int[size5];




    // Store the starting time
    double duration;
    clock_t startTime = clock();
    // Code block


    //For N = 10
    //maxSubSum1(array1,size1);              //0.004 miliseconds.
    //maxSubSum2(array1,size1);              //0.002 miliseconds.
    //maxSubSum3(array1,size1);              //0.003 miliseconds.
    //maxSubSum4(array1,size1);              //0.002 miliseconds.


    //For N = 100
    //maxSubSum1(array2,size2);              //0.546 miliseconds.
    //maxSubSum2(array2,size2);              //0.018 miliseconds.
    //maxSubSum3(array2,size2);              //0.004 miliseconds.
    //maxSubSum4(array2,size2);              //0.003 miliseconds.


    //For N = 1000
    //maxSubSum1(array3,size3);              //465.739 miliseconds.
    //maxSubSum2(array3,size3);              //1.803 miliseconds.
    //maxSubSum3(array3,size3);              //0.081 miliseconds.
    //maxSubSum4(array3,size3);              //0.007 miliseconds.


    //For N = 10000
    //maxSubSum1(array4,size4);              // NA
    //maxSubSum2(array4,size4);              //12756.5 miliseconds.
    //maxSubSum3(array4,size4);              //7.547 miliseconds.
    //maxSubSum4(array4,size4);              //0.514 miliseconds.


    //For N = 100000
    //maxSubSum1(array5,size5);              // NA
    //maxSubSum2(array5,size5);              // NA.
    //maxSubSum3(array5,size5);              //88.434 miliseconds.
    //maxSubSum4(array5,size5);              //5.081 miliseconds.


    //Compute the number of milliseconds that passed since the starting time
    duration = 1000 * double( clock() - startTime ) / CLOCKS_PER_SEC;
    cout << "Execution took " << duration << " milliseconds." << endl;


    return 0;
}
```

Algorithm Table

Input Size	O(N^3)	O(N^2)	O(N logN)	O(N)
N = 10	0.0040	0.0020	0.0030	0.0020
N = 100	0.5460	0.0180	0.0040	0.0030
N = 1000	465.739	1.803	0.0810	0.0070
N = 10000	NA	12756.5	7.547	0.5140
N = 100000	NA	NA	88.434	5.081

Obtained Results (10-100)



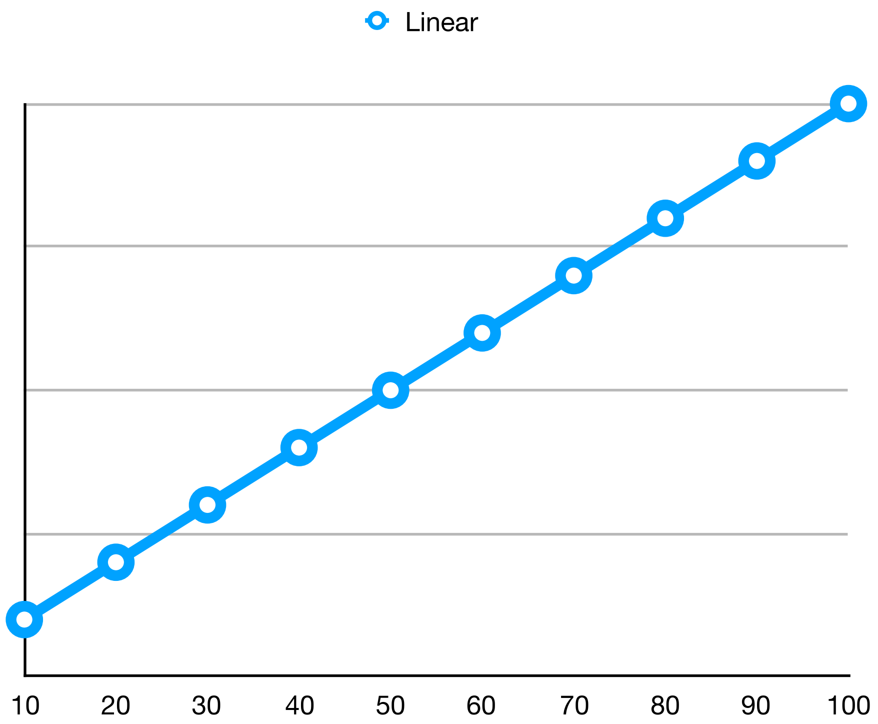


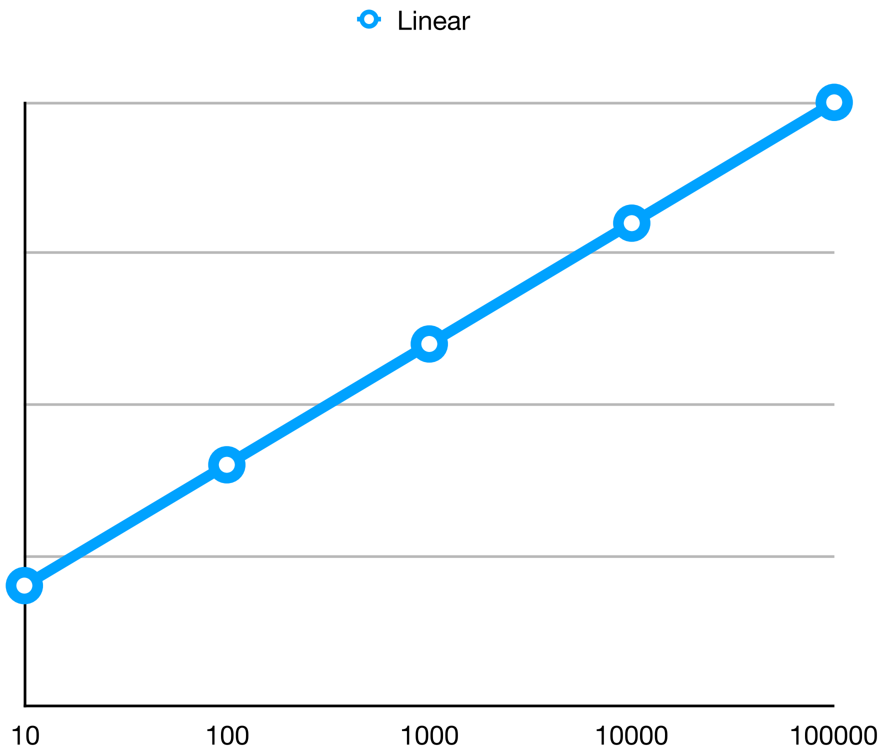
Specification of the Computer that used to obtain these execution times

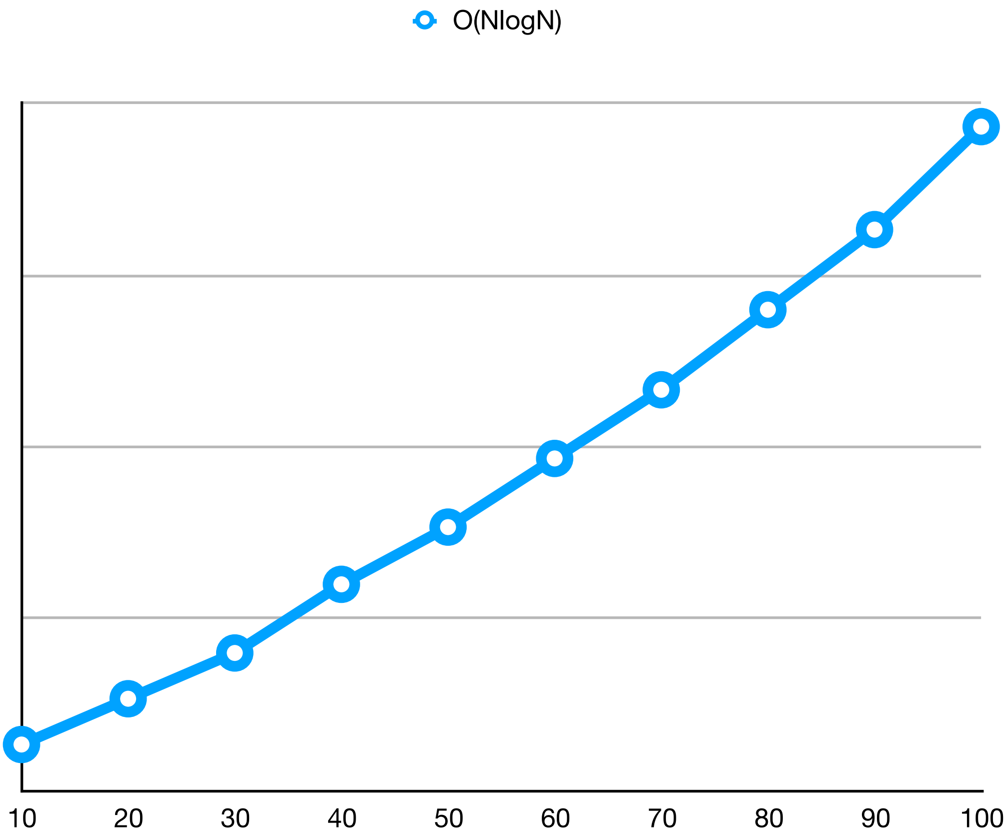
Hardware Overview:

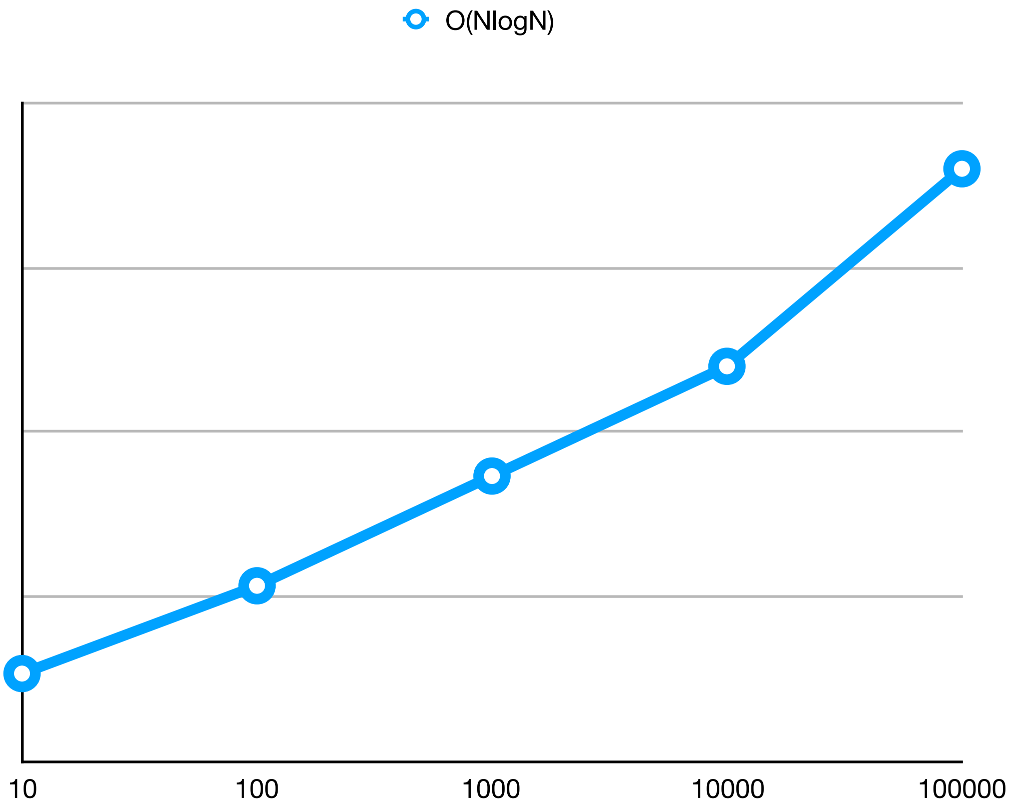
Model Name:	MacBook Pro
Model Identifier:	MacBookPro13,3
Processor Name:	Intel Core i7
Processor Speed:	2,6 GHz
Number of Processors:	1
Total Number of Cores:	4
L2 Cache (per Core):	256 KB
L3 Cache:	6 MB
Memory:	16 GB

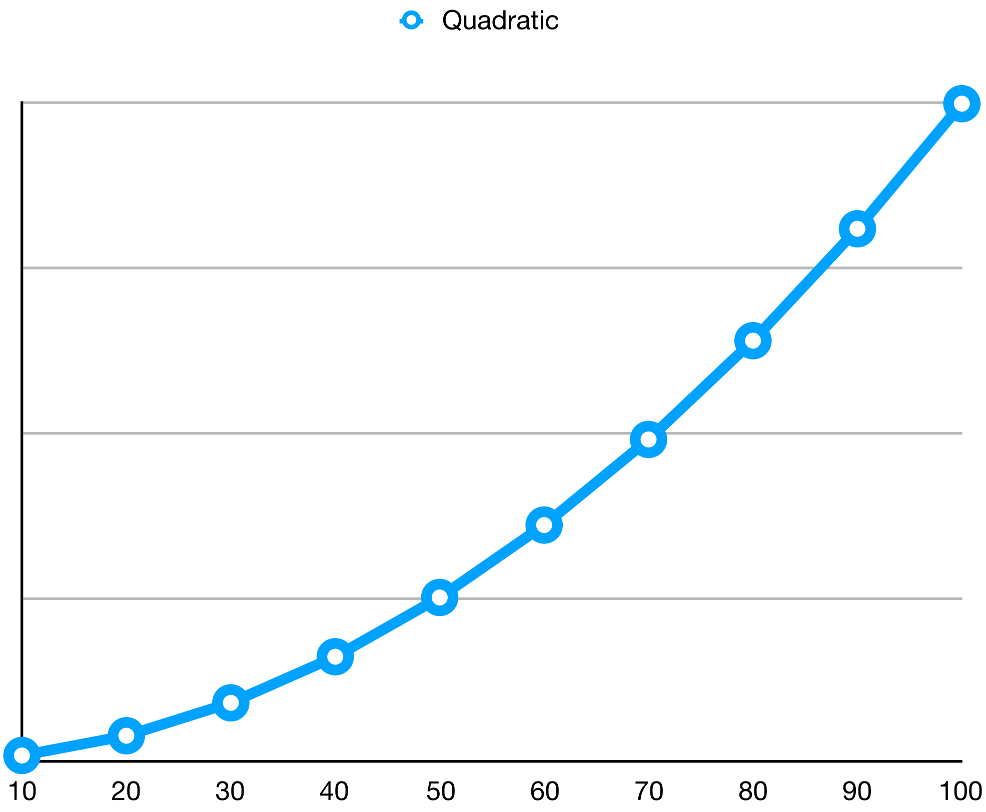
The expected growth rates that obtained from theoretical analysis

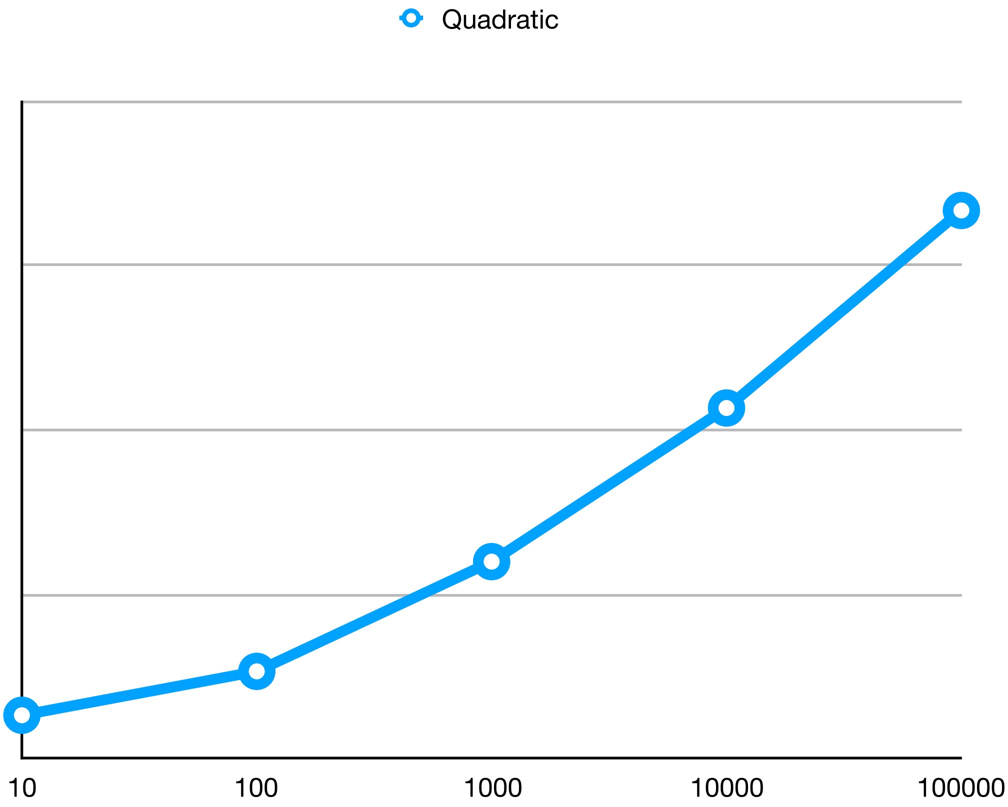


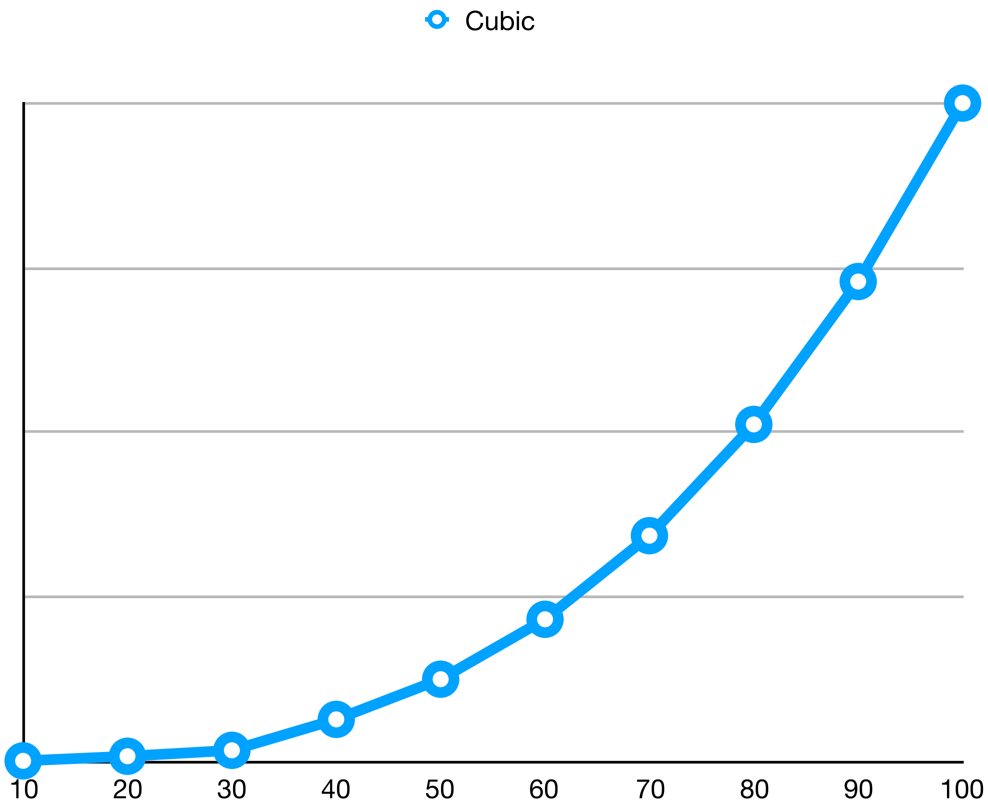


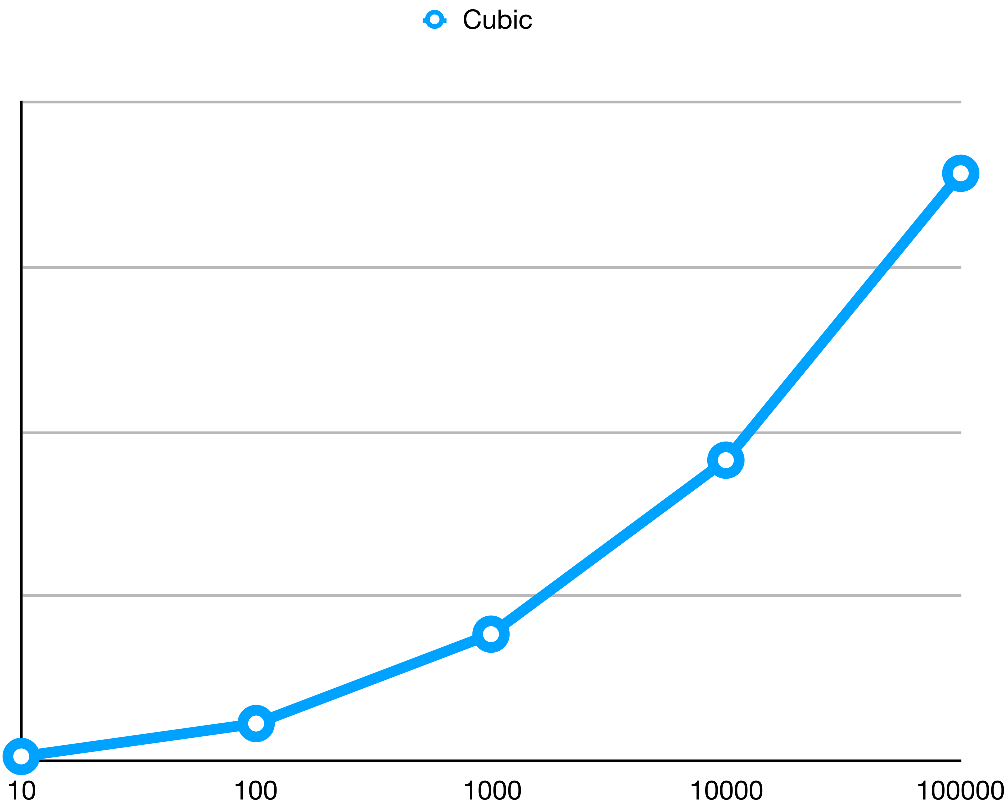

O(NlogN)
10
20
30
40
50
60
70
80
90
100


O(NlogN)
10
100
1000
10000
100000


Quadratic
10
20
30
40
50
60
70
80
90
100


Quadratic
10
100
1000
10000
100000


Cubic
10
20
30
40
50
60
70
80
90
100


Cubic
10
100
1000
10000
100000

Comparison of the expected growth rates and the obtained results

The expected growth which that obtained from theoretical analysis and the growth rates which are obtained from the results have some major differences. The first and the main reason of these difference is the processor speed, and the other opened applications on the computer which can cause a decrement to the speed. Normally the rate of the linear growth should be like a line. On the other hand the growth rates of O(NlogN), quadratic, and cubic are exponential, and their growth speeds are respectively cubic, quadratic, and O(NlogN). As we look at the inputs (N) on the results of these four functions detailed we can see errors. But if we look at the plot in general the growth rates are similar to the growth that we obtained from the theoretical analysis. The speed of the processor can cause bigger errors when the inputs size are closer to each other like in figure 1(10 to 100) than when the inputs size are more far like in figure 2 (10 to 100000). So when the inputs size are increased to draw a graph on the growth rates, the error rate is decreasing. So the relation of the growth rate and the input size (N) is inverse proportion.