



**Bharatiya Vidya Bhavan's**  
**SARDAR PATEL INSTITUTE OF TECHNOLOGY**

(Autonomous Institute Affiliated to University of Mumbai)

Munshi Nagar, Andheri (W), Mumbai – 400 058.

Department of Master of Computer Applications

<b>Experiment</b>	0
<b>Aim</b>	To implement the various functions e.g. linear, non-linear, quadratic, exponential etc.
<b>Objective</b>	To learn about time complexities of various types of functions, implement them and compare them on a graph.
<b>Name</b>	Durgesh Mandge
<b>UCID</b>	2023510032
<b>Class</b>	FYMCA - SEM II
<b>Batch</b>	B (2023-2025)

## Program

```
public class DurgeshFunctionTable {

    public static void main(String[] args) {

        double[][] functionValues =
generateFunctionValues(); printTable(functionValues);

    }

    private static double[][] generateFunctionValues() {

        double[][] values = new double[11][101];

        for (int n = 0; n <= 100; n++) {

            values[0][n] = Math.pow(n, 3); // n^3

            values[1][n] = n; // n

            values[2][n] = n * Math.log(n); // n log n
values[3][n] = Math.log(Math.log(n)); //

            log(log(n))

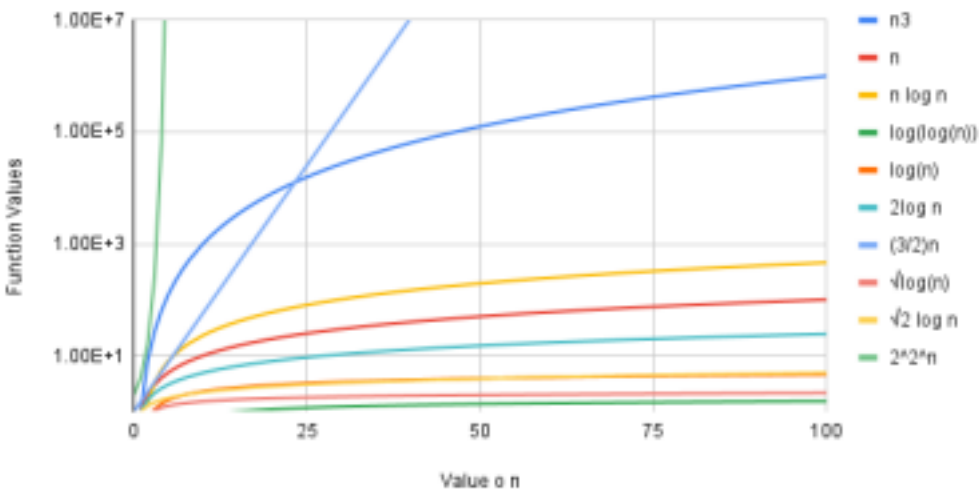
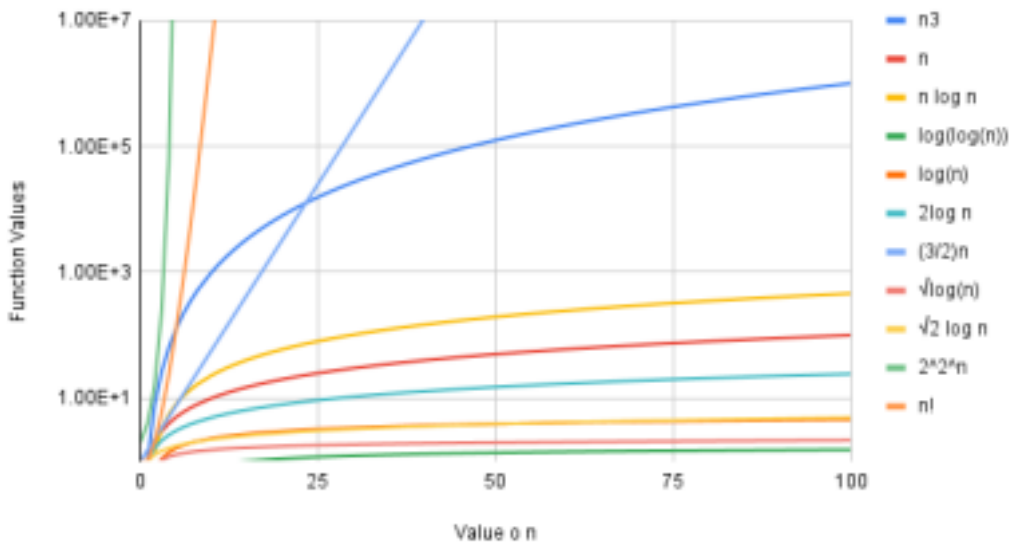
            values[4][n] = Math.log(n); // log(n)

            values[5][n] = Math.pow(2, Math.log(n));
//2^(log(n)) values[6][n] = Math.pow(1.5, n); // (3/2)^n
values[7][n] = Math.sqrt(Math.log(n)); //sqrt(log(n))
values[8][n] = Math.pow(Math.sqrt(2), Math.log(n));

            // (sqrt(2))^log(n)
```

```
        values[9][n] = Math.pow(2, Math.pow(2, n));  
        //2^(2^n) values[10][n] = factorial(n); // n!  
  
    }  
  
    return values;  
}
```

	<pre>    }      private static long factorial(int n) {          if (n == 0    n == 1) {              return 1;          } else {              return n * factorial(n - 1);          }      }      private static void printTable(double[][] values) {          System.out.println("n\t n^3\t n\t logn\t log(log(n))\t log(n)\t 2^(log(n))\t (3/2)^n\t sqrt(log(n))\t (sqrt(2))^log(n)\t 2^(2^n)\t n!");          for (int n = 0; n &lt;= 100; n++) {              System.out.print(n + "\t");              for (int i = 0; i &lt; 11; i++) {                  System.out.print(values[i][n] + "\t");              }              System.out.println();          }      }  }</pre>
Output	
Graph	1. Graph without N! function

	<div><p>Function Graph</p><p>Without N!</p></div> <p>2. Graph with N! function</p> <div><p>Function Graph</p></div>
Conclusion	<p>Presenting the results in a table allowed us to observe patterns and trends, providing a clearer understanding of the functions' behavior. Additionally, we created graphs to enhance our visual comprehension, further deepening our insights into the mathematical relationships.</p>

