Name: Rishi Tiku

Class: SE DS

Subject: Design & Analysis of Algorithms Lab

Experiment: 0

UID: 2021700067

AIM: – To implement the various functions e.g. linear, non-linear, quadratic, exponential etc.

#CODE

```c
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

//helper functions
double fact(double n)//returns factorial
{
    double mult = 1;
    for(;n>=1;n--)
    {
        mult*=n;
    }
    return mult;
}

//actual functions
double fn1(double n)// (3/2)^n
{
    double a = pow((double)3/2, n);
    return a;
}

double fn2(double n)// n^3
{
    return pow(n, 3);
}

double fn3(double n)//log^2 n (base 2)
```

```
{
    return pow(log2(n), 2);
}

double fn4(double n)//log n! (base 2)
{
    return log2(fact(n));
}

double fn5(double n)//2^(2^n)
{
    return pow(2, pow(2, n));
}

double fn6(double n)//n ^ (1/log n) (base 2)
{
    return pow(n, (1/log2(n)));
}

double fn7(double n)//ln (ln n)
{
    return log(log(n));
}

double fn8(double n)//log n (base 2)
{
    return log2(n);
}

double fn9(double n)//n . 2^n
{
    return n * pow(2, n);
}

double fn10(double n)//n^(log log n) (base 2)
{
    return pow(n, log2(log2(n)));
}

void table(double (*f)(double) )
{
    printf("%p\n", f);
```

```
    for ( int ctr = 0 ; ctr <=100 ; ctr+=10 )
    {
        double n = (*f)(ctr);
        printf("|\t%d\t|\t%lf\n",ctr, n);
    }
    printf("\n");
}


void main()
{

    printf("Function : 1 : (3/2)^n\n");
    table(fn1);
    printf("Function : 2 : n^3\n");
    table(fn2);
    printf("Function : 3 : log^2 n (base 2)\n");
    table(fn3);
    printf("Function : 4 : log n! (base 2)\n");
    table(fn4);
    printf("Function : 5 : 2^(2^n)\n");
    table(fn5);
    printf("Function : 6 : n ^ (1/log n) (base 2)\n");
    table(fn6);
    printf("Function : 7 : ln (ln n)\n");
    table(fn7);
    printf("Function : 8 : log n (base 2)\n");
    table(fn8);
    printf("Function : 9 : n . 2^n\n");
    table(fn9);
    printf("Function : 10 : n^(log log n) (base 2)\n");
    table(fn10);
}
```

#RESULT

```
Function : 1 : (3/2)^n
0040144D
|       0       |       1.000000
|       10      |       57.665039
|       20      |       3325.256730
|       30      |       191751.059233
|       40      |       11057332.320940
|       50      |       637621500.214050
|       60      |       36768468716.933022
|       70      |       2120255184830.252000
|       80      |       122264598055704.640000
|       90      |       7050392822843069.000000
|       100     |       406561177535215230.000000

Function : 2 : n^3
0040147C
|       0       |       0.000000
|       10      |       1000.000000
|       20      |       8000.000000
|       30      |       27000.000000
|       40      |       64000.000000
|       50      |       125000.000000
|       60      |       216000.000000
|       70      |       343000.000000
|       80      |       512000.000000
|       90      |       729000.000000
|       100     |       1000000.000000

Function : 3 : log^2 n (base 2)
004014A5
|       0       |       1.#INF00
|       10      |       11.035206
|       20      |       18.679062
|       30      |       24.077575
|       40      |       28.322919
|       50      |       31.853113
|       60      |       34.891357
|       70      |       37.568110
|       80      |       39.966775
|       90      |       42.144157
|       100     |       44.140825
```

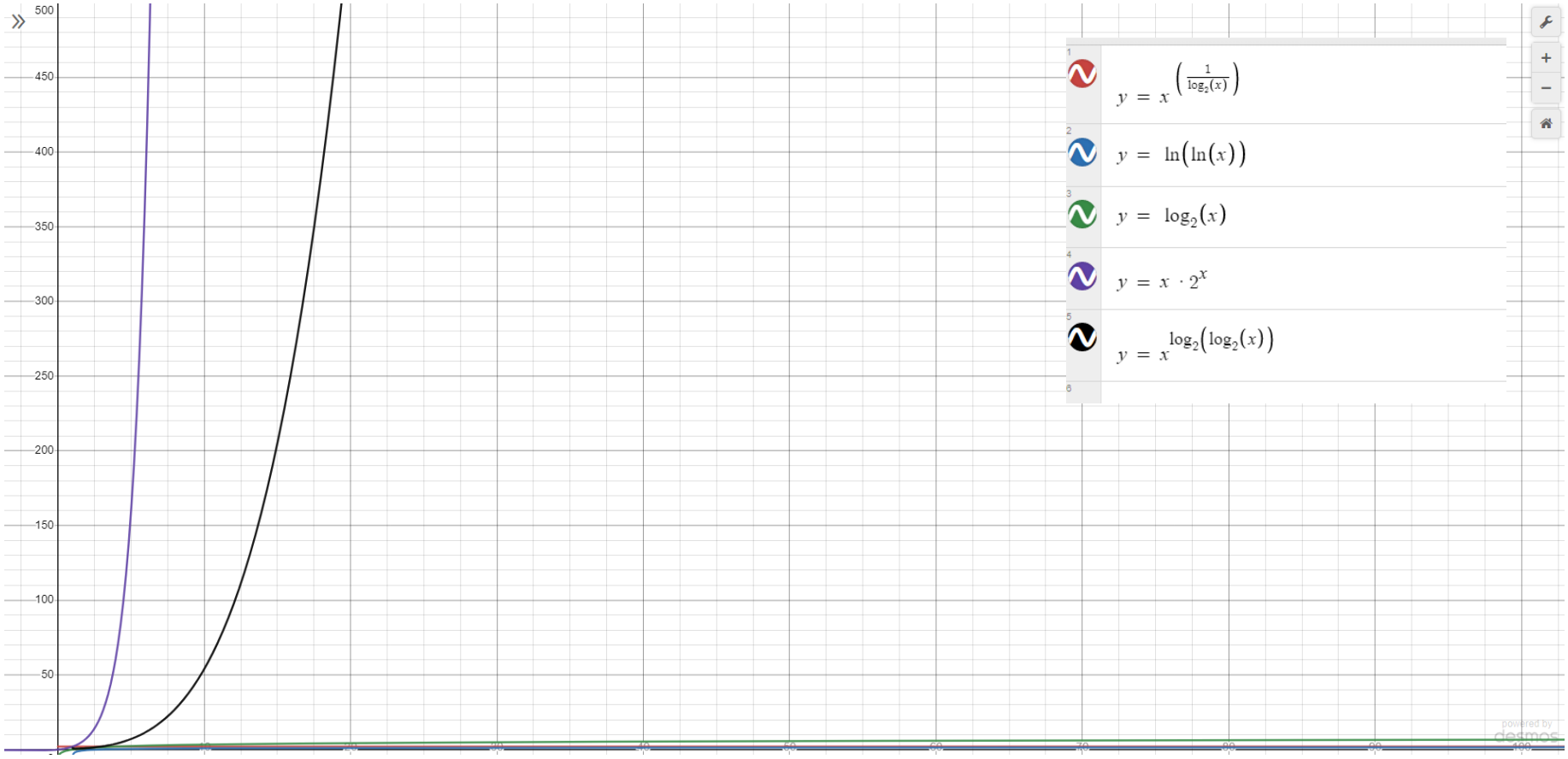
```
Function : 4 : log n! (base 2)
004014D6
|       0       |       0.000000
|       10      |       21.791061
|       20      |       61.077384
|       30      |       107.709067
|       40      |       159.159040
|       50      |       214.208138
|       60      |       272.132930
|       70      |       332.453265
|       80      |       394.826859
|       90      |       458.997235
|       100     |       524.764993

Function : 5 : 2^(2^n)
004014FD
|       0       |       2.000000
|       10      |       1.#INF00
|       20      |       1.#INF00
|       30      |       1.#INF00
|       40      |       1.#INF00
|       50      |       1.#INF00
|       60      |       1.#INF00
|       70      |       1.#INF00
|       80      |       1.#INF00
|       90      |       1.#INF00
|       100     |       1.#INF00

Function : 6 : n ^ (1/log n) (base 2)
00401538
|       0       |       1.000000
|       10      |       2.000000
|       20      |       2.000000
|       30      |       2.000000
|       40      |       2.000000
|       50      |       2.000000
|       60      |       2.000000
|       70      |       2.000000
|       80      |       2.000000
|       90      |       2.000000
|       100     |       2.000000
```

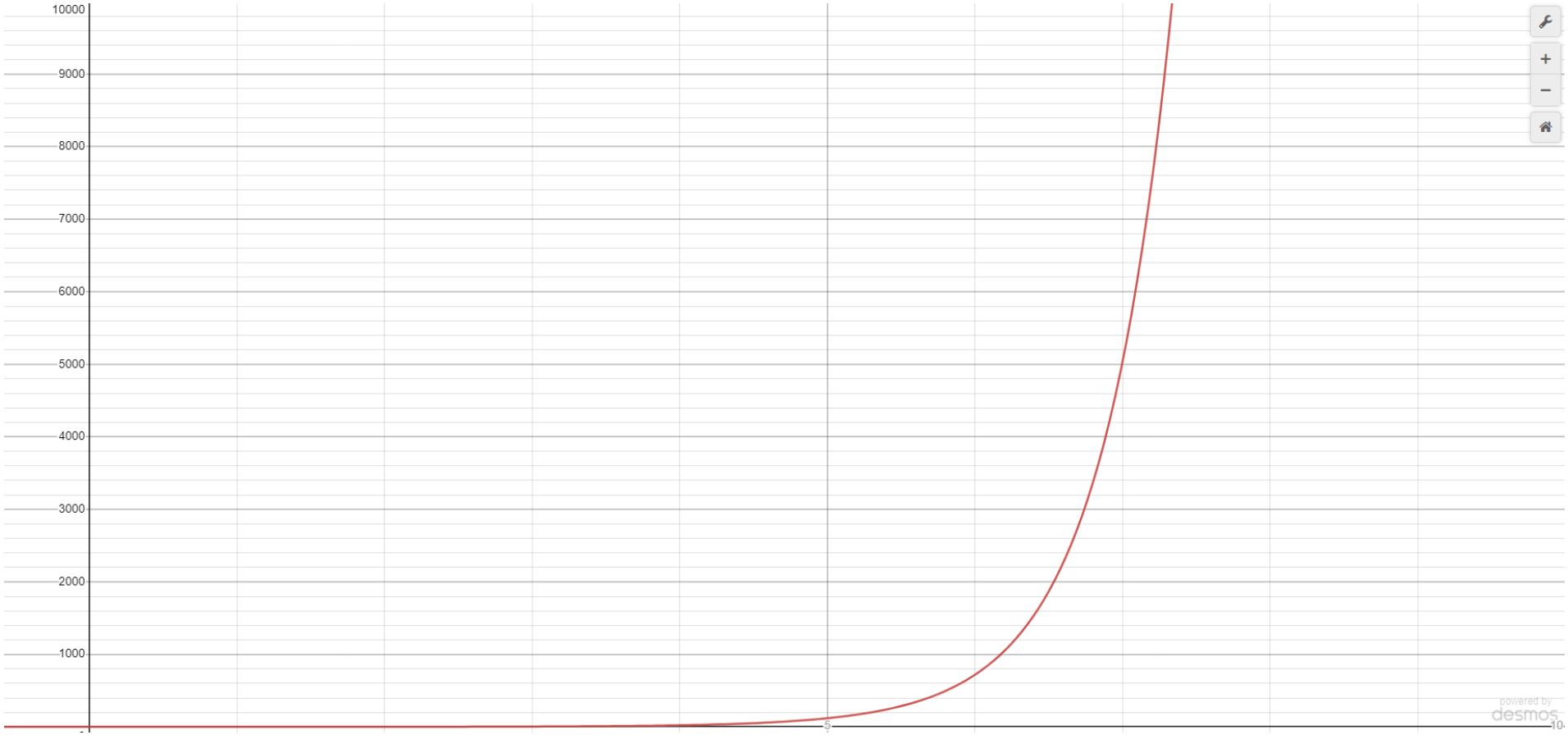
```
Function : 7 : ln (ln n)
0040156A
|       0       |       -1.#IND00
|       10      |       0.834032
|       20      |       1.097189
|       30      |       1.224128
|       40      |       1.305323
|       50      |       1.364055
|       60      |       1.409607
|       70      |       1.446565
|       80      |       1.477511
|       90      |       1.504035
|       100     |       1.527180

Function : 8 : log n (base 2)
00401591
|       0       |       -1.#INF00
|       10      |       3.321928
|       20      |       4.321928
|       30      |       4.906891
|       40      |       5.321928
|       50      |       5.643856
|       60      |       5.906891
|       70      |       6.129283
|       80      |       6.321928
|       90      |       6.491853
|       100     |       6.643856

Function : 9 : n . 2^n
004015B0
|       0       |       0.000000
|       10      |       10240.000000
|       20      |       20971520.000000
|       30      |       32212254720.000000
|       40      |       43980465111040.000000
|       50      |       56294995342131200.000000
|       60      |       69175290276410819000.000000
|       70      |       82641413450218791000000.000000
|       80      |       96714065569170334000000000.000000
|       90      |       111414603535684220000000000000.000000
|       100     |       126765060022822940000000000000000.000000
```

```
Function : 10 : n^(log log n) (base 2)
004015DC
|       0       |       -1.#IND00
|       10      |       53.953652
|       20      |       558.923805
|       30      |       2453.077703
|       40      |       7312.856023
|       50      |       17449.641770
|       60      |       36002.511074
|       70      |       67028.075382
|       80      |       115588.141769
|       90      |       187835.707195
|       100     |       291099.655375
```

#GRAPHS


$y = \left(\frac{3}{2}\right)^x$
$y = x^3$
$y = \left(\frac{\log(x)}{\log 2}\right)^2$
$y = \frac{\log(x)!}{\log 2}$
$y = 2^{2^x}$
500
450
400
350
300
250
200
150
100
50
10
20
30
40
50
60
70
80
90
100


500
450
400
350
300
250
200
150
100
50
$y = x^{\left(\frac{1}{\log_2(x)}\right)}$
$y = \ln(\ln(x))$
$y = \log_2(x)$
$y = x \cdot 2^x$
$y = x^{\log_2(\log_2(x))}$
powered by desmos


10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
5
10
powered by
desmos

1) The domain taken is from X = 0 to X = 100, and the codomain from Y = 0 to Y = 500 for graphs 1 and 2
2) The domain taken is from X = 0 to X = 10, and the codomain from Y = 0 to Y = 10000 for graph 3 which is n!

#OBSERVATIONS

1) Some functions rise more steeply than others. Y = 2^2^n asymptotes to infinity at n ~ 7.5.
2) On the other hand, functions like ln(ln(x)) rise slowly. Here, this function asymptotes at around Y ~ 10.
3) In my opinion, this is an indication that some functions, here, that represent the time taken for completing certain operations, are dependent on the number of operations done. As each algorithm shall have a mathematical function associated with it, which tells us about its time taken, I understand some algorithms are better suited for small number of operations and some work better for a larger number. Their time taken varies on the number of operations.

#RESULT

This Lab activity helped me with intuitive and graphical understanding of the different mathematical functions.