Abe Mathew Mankavil
UTEID: amm23896
CS313E (51130)
2/22/22

Assignment 4

Task 1:

In the first plot where the x-axis is limited to 5, the linear function(seen in red) grows the fastest. The exponential function(seen in green), where n is taken to the power of 3.5, grows the slowest. And lastly the exponential function(seen in blue), where n is taken to the power of 2.1, grows at a rate between the other exponential function and the linear function. In the second plot where the x-axis is limited to 15, the lower power exponential function(green) crosses the linear function and becomes the fastest growing of the 3 functions. The higher power exponential function(blue) appears to still be the slowest growing function and the linear function(red) grows at a rate between the two exponential functions. In the third plot where the x-axis is limited to 50, the higher power exponential function(blue) grows substantially faster than the other two functions. The lower power exponential function(green) follows as the next highest growing function, and lastly the linear function(red) grows the slowest in this frame.

Assignment4

February 21, 2022

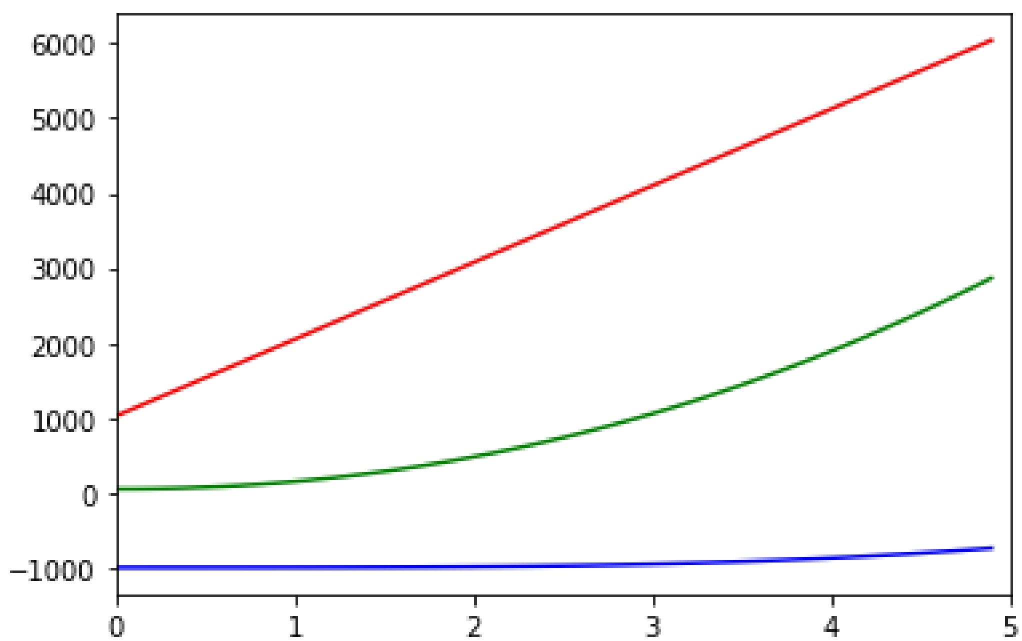
[1]:
```
import math
import numpy as np
import matplotlib.pyplot as plt

msize = 5

n = np.arange(0, msize, 0.1)

plt.plot(n, (2**10)*(n) + (2**10) , 'red', n, n**3.5 - 1000, 'blue', n,␣
↪100*(n**2.1) + 50, 'green')

plt.xlim(0, msize)
plt.rcParams["figure.figsize"] = (7,7)
plt.show()
```



[2]:

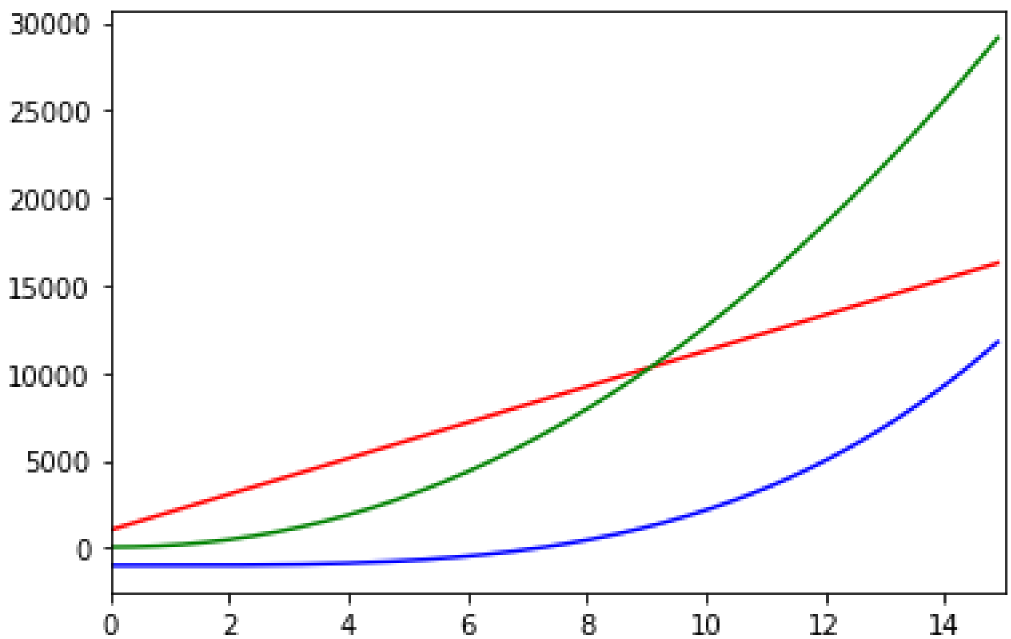
```
import math
import numpy as np
import matplotlib.pyplot as plt

msize = 15

n = np.arange(0, msize, 0.1)

plt.plot(n, (2**10)*(n) + (2**10) , 'red', n, n**3.5 - 1000, 'blue', n,␣
↪100*(n**2.1) + 50, 'green')

plt.xlim(0, msize)
plt.rcParams["figure.figsize"] = (7,7)
plt.show()
```



[3]:

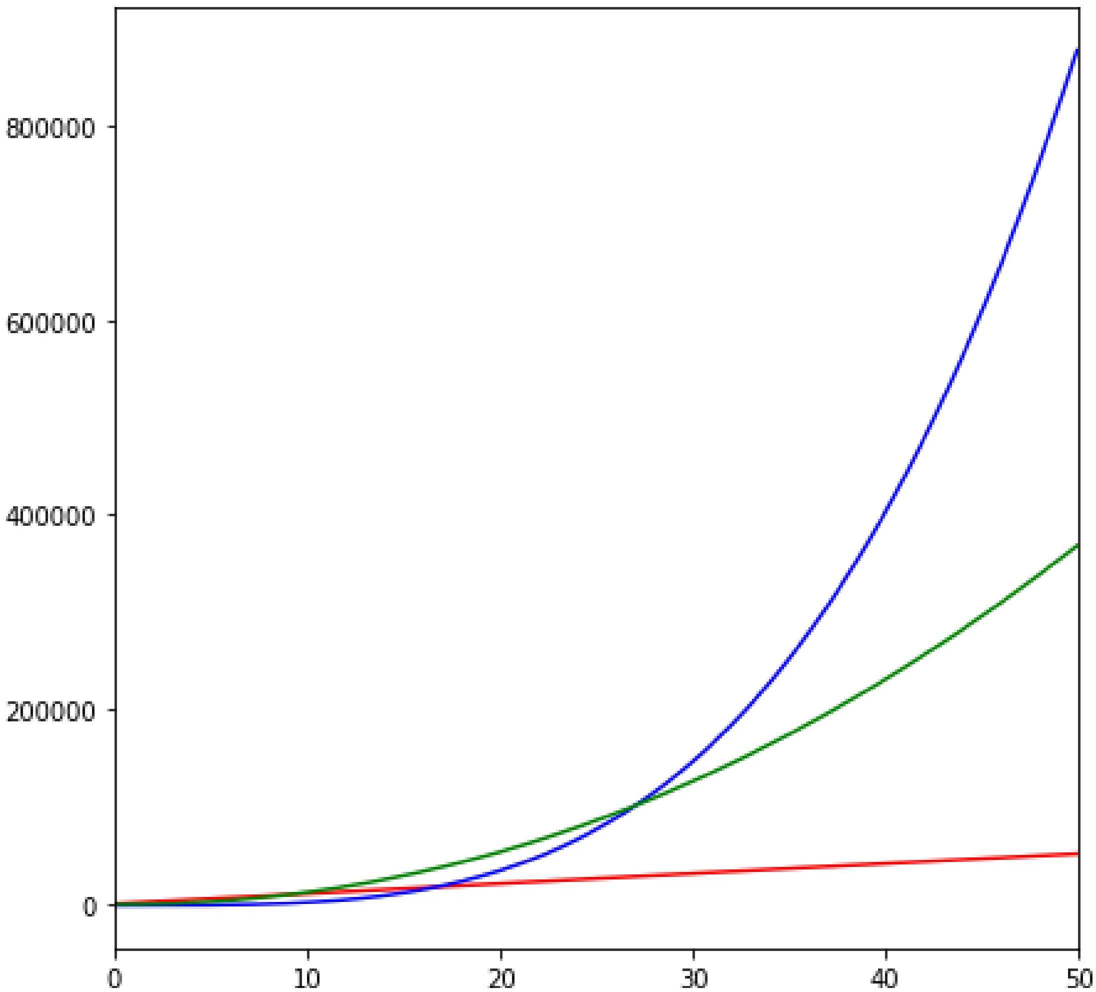
```
import math
import numpy as np
import matplotlib.pyplot as plt

msize = 50

n = np.arange(0, msize, 0.1)

plt.plot(n, (2**10)*(n) + (2**10) , 'red', n, n**3.5 - 1000, 'blue', n,␣
↪100*(n**2.1) + 50, 'green')
```

```
plt.xlim(0, msize)
plt.rcParams["figure.figsize"] = (7,7)
plt.show()
```



Task 2:

a) $2^{(n+1.3)} = O(2^n) \rightarrow$ Yes

$f(n) \leq C \times g(n)$; let $n_0 = 1$

$2^{n+1.3} \leq C \cdot g(n)$

$2^n (2^{1.3}) \leq C \cdot 2^n$

Therefore $\forall n > 1$ and $C \geq 2^{1.3}$, $C(2^n) \geq 2^{n+1.3}$.

Thus $F(n) = 2^{n+1.3} = O(2^n)$

b) $3^{2n} = O(3^n) \rightarrow$ No

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{3^{2n}}{3^n} = \lim_{n \to \infty} 3^n = \infty$$

Therefore $3^{2n} \neq O(3^n)$

Task 3:

I) $f(n) = (4n)^{150} + (2n+1024)^{400}$

$g(n) = 20n^{400} + (n+1024)^{200}$

If we remove all constants added + constant coefficients we are left w/ the equations:

$f(n) = n^{150} + n^{400}$

$g(n) = n^{400} + n^{200}$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^{150}+n^{400}}{n^{200}+n^{400}} \cdot \frac{(1/n^{400})}{(1/n^{400})} = \lim_{n\to\infty} \frac{\frac{1}{n^{250}}+1}{\frac{1}{n^{200}}+1} = \frac{1}{1} = 1$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \neq \infty \rightarrow \therefore f(n) = O(g(n)) ✓$$

II) $f(n) = (4^n)(n^{1.4})$

$g(n) = (3.99^n)(n^{200})$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \approx \lim_{n\to\infty} \frac{1.0025^n}{n^{198.6}} \xrightarrow[\text{198 times}]{\text{take l'Hopitals}} \lim_{n\to\infty} \frac{C_1(1.0025^n)(n^{0.4})}{C_2}$$

$$= \infty$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty, \therefore f(n) \neq O(g(n))$$

III) $f(n) = 2^{\log(n)} = n$

$g(n) = n^{1024}$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{n}{n^{1024}} = \frac{1}{n^{1023}} = 0 \rightarrow \lim_{n\to\infty} \frac{f(n)}{g(n)} \neq \infty, \therefore f(n) = O(g(n)) ✓$$

Scanned with CamScanner

Task 4:

Line 1 ⇒ C_1
Line 2 ⇒ C_2 x $(n+1)$ *Loop iterates through n, while also checking an extra last time
Line 3 ⇒ C_3 x n *Executed the number of times as the while loop is entered
Line 4 ⇒ C_4 x n *See Above Line
Line 6 ⇒ C_5
Line 7 ⇒ C_6 x n
Line 8 ⇒ C_7 x $(n^2 + C_8)$ *Inner loop runs n times for each pass of the for-loop
Line 9 ⇒ C_9 x $(n^2 + C_{10})$ *Executed the number of times as the while loop is entered
Line 10 ⇒ C_{11} x $(n^2 + C_{10})$ *See Above Line

$T(n) = C_{11}(n^2 + C_{10}) + C_9(n^2 + C_{10}) + C_7(n^2 + C_8) + C_6 n + C_5 + C_4 n + C_3 n + C_2(n+1)$
$T(n) = an^2 + bn + c;$ where $a,b,$ and c are a sum of constants
$T(n) = O(n^2)$

Task 5:

Line 1 ⇒ C_1
Line 2 ⇒ C_2 x 1
Line 3 ⇒ C_3 x $(n + 1)$ *Inner loop runs same # of times as the outer loop iterates through n
Line 4 ⇒ C_4 x $(n^3 + n^2 + C_5)$ *contents of inner loop is run $(n \cdot \sum_{k=0}^{n} k) = (n \cdot \frac{n(n+1)}{2})$

$T(n) = C_4$ x $(n^3 + n^2 + C_5) + C_3$ x $(n + 1) + C_2$ x $1 + C_1$
$T(n) = an^3 + bn^2 + cn + d;$ where $a,b,c,$ and d are a sum of constants
$T(n) = O(n^3)$