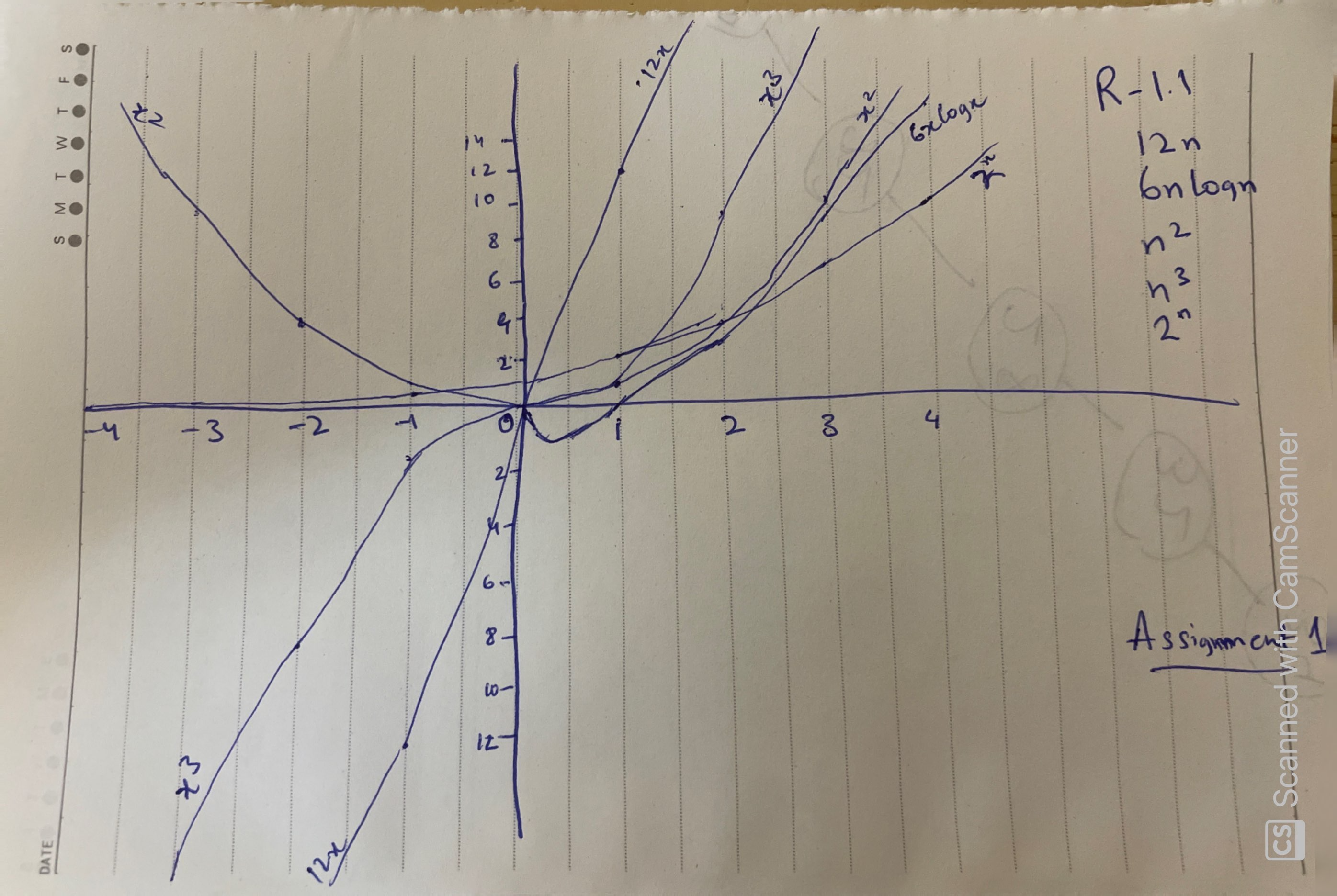
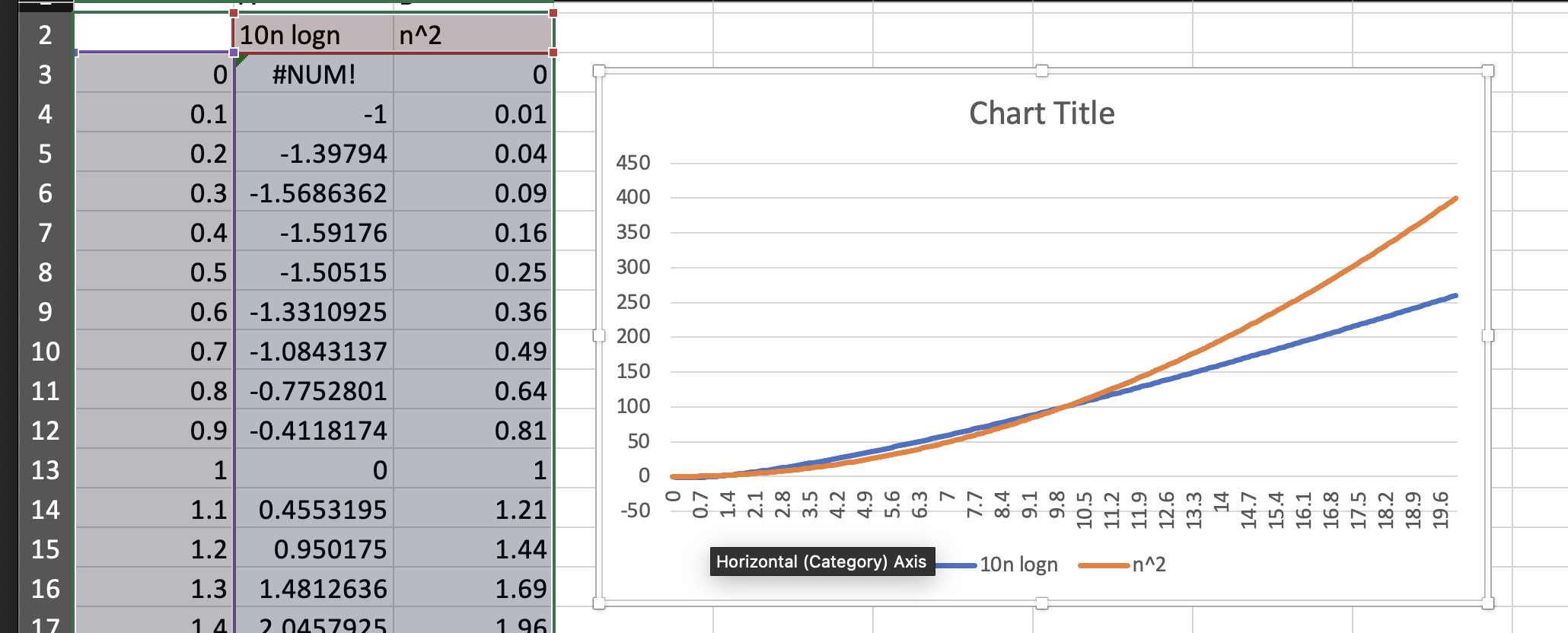
Assignment 1

**R-1.1 Graph the functions 12n, 6n log n, n2, n3, and 2n using logarithmic scale for the x- and y-axes; that is, if the function value f(n) is y, plot this as a point with x-coordinate at log n and y-coordinate at log y.**



**R-1.2 Algorithm A uses 10n log n operations, while algorithm B uses n2 operations. Determine the value n0 such that A is better than B for n ≥ n0.**

For n0 is 1 and 10 , A is better than B



**R-1.6 Order the following list of functions by the big-O notation.**

**n log n log log n , 1/n 4n3/2**

**5n 2n log2 n, 2n 4n**

**n3 n2 log n 4log n √n**

|  |
| --- |
| 1. log log n 2. 1/n 3. √n 4. 4log n 5. n log n 6. 2n log2 n 7. 5n 8. n2 log n 9. 4n3/2 10. n3 11. 2n 12. 4n |

R-1.10 Give a big-O characterization, in terms of n, of the running time of the Loop1 method below:

|  |  |
| --- | --- |
| Algorithm Loop1(n)  s ← 0  for I ← 1 to n do  s ← s + i | O(1)  O(n)  O(n)  Total running time = O(n) |

R-1.14 Perform a similar analysis for method Loop5 below:

|  |  |
| --- | --- |
| Algorithm Loop5(n)  s ← 0  for I ← 1 to n2 do  for j ← 1 to i do  s ← s + i | O(1)  O(n2)  n2 (n2+1)/2  n2 (n2+1)/2  Total running time = O(n4) |

**Prove:  
logbxa =alogbx**

Here we assume :

**y=logbxa**

So as per Log rule :

***bx=a <=> logba=x***

**y=logbxa**  ***<=>* if by = xa**

Using the laws of exponents, we can rewrite

**(by) 1/a = (xa) 1/a *<=>*  b y/a = x**

Now we can convert to logarithm using above logarithm rule

**b y/a = x *<=> logbx = y/a <=> alogbx = y***

Now we can put in y.

**alogbx = y <=> logbxa =alogbx**