NAME \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Write the following functions in order on the lines below so that the fastest growing function is first and the slowest growing function is last. If two functions grow at the same rate, write them on the same line. You may not need to use all lines.

, , , , , , , , , , , , , , , , , , .

1. \_\_\_\_\_\_\_\_\_\_\_2^2^n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n^n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n!\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2^nlgn\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3^n\_\_\_2^n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n^3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
7. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, n^2 n^2 + n \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_nlog(n)\_
9. 3n+1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n \_\_\_\_
10. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n^.0001\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
11. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_lg^2 (n)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
12. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
13. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2 lg(n)\_log3(n)\_log(n)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
14. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ lg lg (n) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
15. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
16. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
17. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
18. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
19. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
20. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
21. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Give the growth rate (order) of the following recurrences. You may assume some reasonable value for  or 

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_1 + 2 + 3 + 4… n = O(n)\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n^log5(5) = n = O(nlg(n))\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_n^(log4(16)) = n^2 \_\_\_\_

\_\_\_\_\_\_\_\_\_\_n^5(25) =< n^3 = (n^3) \_\_\_\_\_

\_\_\_\_\_\_\_\_2^n + 2^n-1 + 2^n-2 + 2^n-3 … = 2^n\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_O(n)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, 

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n^1 >c \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n^log2(2) <2^n = O(2^n)\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_n^log2(4) = n^2>lg(n) = O(n^2)\_\_\_\_\_

3. Prove that the function  grows faster than  where , .

N = 2 , c = 1

A = 1

A = 2

N^2 > ln(2)

N^3 > ln(2) -> proven

Lim n->infinity ( n^a / ln(n))

Lim ((a-1) \* n^a \* n ) = infinity. Therefore n^a is faster than ln(n)

4. Prove that the function  grows slower than .

Let n0 = 2, c= 1

2^2 = 4

3^2 = 9

2^n < 3^n whenever n >n0 therefore 2^n E O(3^N)

5. Define the following, 

O(g) says f(n) is O(g(n)) if n>n0 and f(n) < c \* g(n) .

6. Which is faster (in terms of time bound) depth first search or breadth first search?

7. Prove that searching an unordered list of  elements requires at least  time regardless of how it works.

Best Case:

e

8. Suppose, for a certain input, Quicksort always places the pivot point in the second position of the list after the Split operation. Prove that sorting this list requires  time.

Quicksort is a sorting algorithm that sorts by splitting up two recursive arrays at a pivot. At that pivot all elements below are less and all elements above are more. During the sorting, each element is compared to the pivot and the pivot location is “moved” such that the two small arrays gain/lose elements. In the wrost case when the pivot is always near the front (2nd element) n-1 comparisons have to be done at each step, but no real progress made because the pivot returns. This means that the loop will do n-1, n-2, n-3 which is overall the sum(i) fmro i->n-1 = n(n-1)/2 = O(n^2).

9. Does the Dijkstra-Prim Minimum Spanning Tree algorithm work for mixed positive and negative edge weights? Why or why not?

10. Use the substitution method to prove that  is 

N^2 + (n-1)^2 + (n-2)^2 +(n-3)^2 => summation of i^2 =( 2n^3 + 3n^2+ n )/ 6 =