These are only sample questions to indicate the types of questions that may be on the midterm. The best way to study is make up your own versions of these questions and try to answer them without any reference to notes or text or internet.

What notation, O( ), Θ( ), Ω( ), **best** describes the relationship between the two functions shown below.

1. log (n^n) is ____ (n^2) a) O( ) b) Θ( ) c) Ω( )

2. As DFS explores the edges of a **directed** graph, list the different types of edges that it may encounter

3. What will be the index of "25" after it is inserted in this MaxHeap using the usual algorithm.

index	1	2	3	4	5	6	7	8	9
value	30	29	23	27	22	19	16	26	18

   a) 2 b) 3 c) 5 d) 8 e) none of these

4. Construct a MaxHeap for 2 7 28 4 30 3 29 using **bottom up construction**? .

5. What is the number of inversions in the following list of rankings? 4 5 7 2 3 1 6

6. What is the value of $\sum_0^{10} 3^i$?

7. Given a set of size 2^n of distinct bitstrings of length n, what is the condition that makes the set a Gray Code.

8. Find the topological sort of the vertices of a DAG using the DFS based algorithm following the usual convention of using alphabetical order to break ties when necessary.

9. Give the recurrence relation that describes the number of instructions for a problem of size n for any of the recursive algorithms discussed in class, homework, labs, etc.

10. Which of the following recurrence relations best describes number of key comparisons that would occur in MergeSort **in the worst case**? Include the work done by the partitioning algorithm.

    a) T(n) = 2*T(n/2) + n b) T(n) = 2*T(n/2) + n/2
    c) T(n) = T(n/2) + n d) T(n) = T(n-1) + n

11. For a given graph which of the following is the order in which vertices are added to the minimum spanning tree using Prim's algorithm starting at vertex A.

12. Fill in the blanks to give the formal mathematical definition of f(n) is in $\Omega(g(n))$.

If f(n) and g(n) are functions from $Z^+ \rightarrow Z^+$. Then f(n) is in $\Omega(g(n)$ if there exists

__________________________ such that_______________________________

$\forall$______________________________.

13. Use back substitution to determine the <u>closed form solution</u> and the computational complexity class for the following recurrence relation. T(n) = 3*T(n/2) + 1 for n>1 and T(1) = 0.

<u>**Show derivation here:**</u>
T(n) = 3*T(n/2) + 1
….

T(n) =

Closed Form Solution (include constants): ______________ **Order of Growth** = θ(_________)

14. What would be the result of applying Lomuto's partitioning algorithm to a set of numbers using a specified number as the partitioning element.
15. What would be the result of applying Hoare's partitioning algorithm to a set of numbers using a specified number as the partitioning element.
16. Given the Master Theorem, Give the complexity class for a function defined by a given recurrence relation and base case.
17. **Licensing strategy** Suppose you own a company that must license n software modules. Since your company has only a limited amount of money to spend each month, you can only purchase one license per month. The costs of the software licenses are all different and are given by $p_1, \dots, p_n$. Unfortunately the cost of all the licenses goes up by a factor of r (r is > 1) each month. Thus the price of a license for the i^{th} product is $p_i * r^m$ after m months.
    a. Specify the ordering that results in the least overall cost.
    b. Prove the correctness of your algorithm using an exchange argument