Design and Analysis of Algorithms: Sheet-1

Note - Refer standard books Ex. **Cormen**, or websites like **geeksforgeeks** or **stackoverflow** for solving problems.

1. What do you understand by Asymptotic notations. Define different Asymptotic notation with examples.

2. What should be time complexity of – for(i=1 to n) {i=i*2}

3. T(n) = {3T(n-1) if n>0, otherwise 1}

4. T(n) = {2T(n-1)-1 if n>0, otherwise 1}

5. What should be time complexity of -

```
int i=1, s=1;
while(s<=n){
  i++; s=s+i;
  printf("#");
}
```

6. Time complexity of -

```
void function(int n){
    int i, count=0;
    for(i=1; i*i<=n; i++)
        count++          }
```

7. Time complexity of -

```
void function(int n){
    int i,j,k, count=0;
    for(i=n/2; i<=n; i++)
        for(j=1; j<=n; j=j*2)
            for(k=1; k<=n; k=k*2)
                count++  }
```

8. Time complexity of -

```
function(int n){
        if(n==1) return;
            for(i=1 to n){
                for(j=1 to n){
                    printf("*");
                }
            }
        function(n-3);
}
```

9. Time complexity of -

```
void function(int n){
    for(i=1 to n){
        for(j=1; j<=n; j=j+i)
            printf("*")
    }
}
```

10. For the functions, n^k and c^n, what is the asymptotic relationship between these functions? Assume that k >= 1 and c > 1 are constants. Find out the value of c and n0 for which relation holds.

11. What is the time complexity of below code and how?

```
void fun(int n){
  int j = 1, i = 0;
  while (i < n){
    i = i + j;
    j++;}}
```

12. Write recurrence relation for the recursive function that prints Fibonacci series. Solve the recurrence relation to get time complexity of the program. What will be the space complexity of this program and why?

13. Write programs which have complexity – n(logn), n^3, log(logn)

14. Solve the following recurrence relation T(n) = T(9n/10) + T(n/10) + cn

15. What is the time complexity of following function fun()?

```
int fun(int n){
  for (int i = 1; i <= n; i++){
    for (int j = 1; j < n; j += i){
      // Some O(1) task
    }}}
```

16. What should be the time complexity of

```
for (int i = 2; i <=n; i = pow(i, k))
{
   // some O(1) expressions or statements
}
```

where, k is a constant.

17. Write a recurrence relation when quick sort repeatedly divides the array in to two parts of 99% and 1%. Derive the time complexity in this case. Show the recursion tree while deriving time complexity and find the difference in heights of both the extreme parts. What do you understand by this analysis?

18. Arrange the following in increasing order of rate of growth:

a) n, n!, logn, loglogn, root(n), log(n!), nlogn, log^2(n), 2^n, 2^(2^n), 4^n, n^2, 100

b) 2(2^n) , 4n , 2n , 1 , log(n) , log(log(n)) , √log(n) , log2n , 2log(n) , n , log(n!) , n! , n2 , nlog(n)

c) 8^(2n) , $\log_2(n)$, $n\log_6(n)$, $n\log_2(n)$, log(n!) , n! , $\log_8(n)$, 96 , 8n2 , 7n3 , 5n

19. Write linear search pseudocode to search an element in a sorted array with minimum comparisons.

20. Write pseudo code for iterative and recursive insertion sort. Insertion sort is called online sorting. Why? What about other sorting algorithms that has been discussed in lectures?

21. Mention the time and space complexity of all the sorting algorithms that has been discussed in lectures.

22. Divide all the sorting algorithms into inplace/stable/online sorting.

23. Write recursive/iterative pseudo code for binary search. What is the Time and Space complexity of Linear and Binary Search (Recursive and Iterative)

24. Write recurrence relation for binary recursive search.

25. Find two indexes such that A[i]+A[j] = K in minimum time complexity.

26. Which sorting is best for practical uses? Explain.

27. What do you mean by number of inversions in an array? Count the number of inversions in Array arr[] = {7, 21, 31, 8, 10, 1, 20, 6, 4, 5} using merge sort.

28. In which cases Quick sort will give the best and the worst case time complexity?

29. Write Recurrence Relation of Merge and Quick sort in best and worst case?

30. Selection sort is not stable by default but can you write a version of stable selection sort.

31. Bubble sort scans whole array even when array is sorted. Can you modify the bubble sort so that it doesn't scan the whole array once it is sorted.

32. Write an algorithm for counting sort and mention its time and space complexity. In which we should use counting sort or in which case the counting sort is efficient?

33. Given an array A of n elements, each of which is an integer in the range [0, n^2]. How do you sort the array in O(n) time? ***Hint- Use Radix Sort***

34. Given an array with n integers each of value less than n^100. Can it be sorted in linear time?

35. Write an algorithms to sort an array of 0's, 1's and 2's in O(n) time complexity.

36. Your computer has a RAM (Physical memory) of 2 GB and you are given an array of 4 GB for sorting. Which algorithm you are going to use for this purpose and Why ? Also explain the concept of External and Internal Sorting.

37. What do you understand by Greedy algorithms? How does Greedy Choice work for activity selection problem when Activities are sorted according to finish time? How to implement algorithm if activities are not sorted by finish time?

38. What is minimum spanning tree (MST)? What are the applications of MST? Write an algorithm to detect cycle in Kruskal and Prim's algorithm. Explain the time complexity of both the MST finding algorithms.

39. Why doesn't Dijkstra's algorithm work for negative weight edges? Can you find out the similarities and differences between implementation and uses of Dijkstra and Prim's algorithm?

40. Given a directed weighted graph. You are also given the shortest path from a source vertex 's' to a destination vertex 't'. Does the shortest path remain same in the modified graph in following cases?
1. If weight of every edge is increased by 10 units.
2. If weight of every edge is multiplied by 10 units.