Data Structures & Algorithms - Interview Questions

1. When you use malloc or new why is memory allocated on the Heap and not the Stack ?

What will be consequence if dynamic memory allocation happens from the Stack ?

2. What is a Segment Tree ? What are the applications of Segment Trees ?

3. Why do we say that searching for an element in a Hash Table can be greater than O(1)

in the worst case ?

4. If you want efficient search which data structure would you use

a. BST

b. Heap

c. Hash Table

Explain your answer

5. How will you prove that the time complexity of Binary Search is O(logn)

6. What is the time complexity of Bubble Sort ? If the array is sorted will the time

complexity be the same ?

7. What is the worst case time complexity of Quick Sort ? Why is it so ? Can you explain

with an example ?

8. Why do we need Postfix and Prefix expressions ? Why can't we use Infix ?

9. How will you determine all the 0 sum subarrays in a array

10. Given 2 arrays array1 and array2 how will you determine if array2 is a subset of

array1. What will be the time complexity of your solution.

11. Explain the concept of Hashing with an example

12. What are the different collision resolution mechanisms

when you use Hashing ?

13. What is Quadratic Probing ? How does it minimize primary clustering ?

14. How do you find the worst case time complexity when you use Chaining as the

collision resolution mechanism ?

15. Compare and Contrast Open addressing schemes Vs Chaining

16. Given the following array elements how would use Heap Sort to sort the array

A = [ 5, 1, 9, 2, 6, 4, 3 ]

17. How would you check if a given array is a Heap ( Max or Min ) ?

18. How would you determine the first non repeating character in a string ?

19. How would you split a sentence into words in C/C++ using strtok ?

20. How would you find missing elements within a range in a array ?

21. How would you check if 2 arrays are equal ?

22. How would you determine the element that occurs with the maximum frequency

in an array ?

23. What is double hashing ? What does problem does it solve ? What are the limitations ?

24. How would you decide on the size of the hash table ?

25. Explain the difference in strategy between pattern matching using algorithms such

as KMP and Trie

26. How is the pattern pre-processed using KMP algorithm ? What is the time complexity

of searching for a string of m characters in a text of n characters ?

27. How would you determine if a number is a power of 2 in an efficient way ?

28. How would you convert the following infix expression to postfix ?

Infix => A + B \* D + (E - F)/G \* M - K

29. What is the issue with a linear queue ? How is that problem overcome in a

circular queue ?

30. Write the enqueue and dequeue functions for a circular queue

Assume the queue size is defined by the macro MAX\_QUEUE\_SIZE

31. Write the push and pop functions for a stack implemented as an array.

Assume the stack size is defined by the macro MAX\_STACK\_SIZE

32. Write a function to insert a string into a trie. Assume that only lower case

English alphabets are used. Please find the structure of the trie node below

analyze the time complexity of the insert function

#define MAX\_ALPHABET\_SIZE 26

struct trienode

{

struct trienode \*children[MAX\_ALPHABET\_SIZE];

boolean end\_of\_word;

};

33. Write a function to search for a string in a trie given the root of the trie

What is the time complexity of searching for a string in a trie ?

34. Derive the time complexity of Binary Search and Quick Sort using

a. Masters Theorem

b. Back Substitution Method

35. How would you delete a node in a singly linked list given only the address of the

node ? You are not given the head pointer of the linked list.

36. Arrange the following time complexities of algorithms in alphabetical order.

Explain your answer

a. O(n)

b. O(logn)

c. O(2 power n)

d. O(n\*n)

e. O(nlogn)

f. O(sqrt(n))

g. O(n factorial)

37. Why do we analyze the time complexities of an algorithm ? Can we not rely on the

execution times ?

38. Explain what do you understand by Dynamic Programming. Explain what is

a. Optimal Sub Structure

b. Overlapping Sub Problems

39. What is the recurrence relation for the following recursive algorithms ?

Explain how you arrived at the recurrence relation

a. Binary Search

b. Fibonacci Series

c. Merge Sort

40. How would you determine the space complexity of a recursive algorithm ?

41. Distinguish between Memoization & Tabulation

42. Analyze the time complexity of the following operations

a. BST to Max Heap

b. Max Heap to BST

43. What is the time complexity of searching for a string in an 2D array that is

sorted row wise as well as within the row. That means if you place all the rows

one after another in order the array is completely sorted

44. Explain how you would search for a string of length m in a text of length n using

a. Brute Force algorithm

b. KMP algorithm

c. Boyer Moore algorithm

d. Rabin Carp algorithm

Analyze the time complexity of each of these algorithms

45. Explain why Huffman algorithm is classified as Greedy. How does it achieve

Data Compression ?

46. How woud you create a

a. A Queue from 2 stacks

b. A Stack from 2 Queues

Analyze the time complexity of Enqueue & Dequeue functions of the former as well

as the Push & Pop functions of the latter

47. Explain the difference between Big Oh, Theta and Omega notations

48. Analyze the space complexities of the following

a. Recursive Fibonacci Algorithm

b. Memoized Fibonacci Algorithm

c. Tabulated Fibonacci Algorithm

49. Explain the following rotations in an AVL Tree with an example

a. LR Rotation

b. RL Rotation

50. What is the time complexity of determining the longest palindromic substring using

a. Brute Force

b. Dynamic Programming

51. What is the time complexity of determing

a. All the substrings in a string

b. All the subseuence in a string

c. To find if the given string is a substring

d. To find if the given string is a subsequence

e. To find if the given string is a palindrome

52. Compare & contrast the following collision resolution techniques

a. Linear Probing

b. Quadratic Probing

c. Chaining

d. Double Hashing

53. We can write program for factorial using iteration as well as recursion. Which approach

is more efficient in time and space ? Explain your answer

54. How can you determine if a number is even or odd without using the modulus operator

55. What are the data structures used to represent graphs ?

56. What would be the time complexity to determine the following

a. if there is an edge between 2 vertices

b. What are all the directly connected vertices from a given vertex

Using

1. Adjacency Matrix

2. Adjacency List

3. Edge List

57. What would be your approach to determine if a given tree is a subtree of another tree

58. What would be the time complexity of determining the Kth largest element in

an unsorted array using

a. Sorting

b. Min Heap

c. Max Heap

d. Quick Select

59. What would be the time complexity of checking if the given binary tree is max

heap using

a. Level Order Traversal

b. Recursive approach

60. Explain what is Topological Sorting