INTERVIEW QUESTIONS ON SEARCHING

1. What is Searching ? Identify Different types of Searching techniques based on different type of data structures.

Ans. When we have data stored in a variable or in a data structure obviously we want to find it later. So, to find that variable we use searching we may do operations on that variable or modify that variable but to do anything we have to find it first that using searching technique.

Searching Techniques :

1. Linear Search : Useful for any type of data structure however it is and any type of data structure it is. But it will entirely **traverse** the data structure. It may start from first or from the last. It's Time Complexity is O(n).

2. Binary Search : Only works when the data structure is **sorted** that too it won't work for linked list data structure. It is based on the principle of Reduce and Conquer. It's time complexity is O(log n).

3. Ternary Search : This is just like binary search but the main difference is binary search divides the array into 2 sub arrays whereas the ternary search divides the array into **3 sub arrays**. It's time complexity is O(log n_3).

4. Fibonacci Search : -

5. Sentinel Search : -

6. Hashing : This search is mainly useful for the map data structures based on the hashing principles like hashmaps, treemaps, etc., As this uses direct accessing of that element it is either O(1) for maps and sets or O(log n) for tree map(for sorting).

7. Indexing : It is mainly used for the tree data structures like Binary search tree, B tree, B+ Tree, red black tree etc., for the searching purpose.

2. What is Linear Search and Identify it's Space complexity and Time complexity of this in different contexts.

Ans. Linear Search : It is one type of searching technique where the entire data structure will be traversed (in worst case). Because in this we check by taking one value and checking whether that matches with our desired value or not. If it matches then our element is found. If not we then take the next value and this process goes on until the end of the array. If the element is not found we return -1 or false indicating that there is no our desired element in the data structure.

Space Complexity : As we are not using any extra space. It is O(1).

Time Complexity :

1. Best Case : If the element is found at very first iteration or on very few iterations then the time complexity will be O(1).

2. Average Case : If the element is found at middle index of the data structure or very near to middle index then it is O(n/2).

3. Worst Case : If the element is found at end of data structure which means entire data structure is traversed so it is O(n).

3. What is Binary Search and how it is different to Linear Search.

Ans. Binary Search : It is one of the efficient searching technique which works on sorted data structures only. But it is mainly popular for its reduce and conquer principle. Because it will first define the boundaries(lower and upper) and then it will calculate the mid value and checks with our desired value if found then element is in the data structure and can break if not found then it divides the array into 2 sub arrays and then checks if desired value is greater than or lesser than mid value. Based on that it will change the upper boundary and lower boundary. And again do the same steps.

How it is different to Linear Search : Linear search works with any type of data structure whereas the binary search works with only **sorted** data. Linear search TC is O(n) in worst case whereas the Binary search TC is O(log n) in worst case. Linear search traverses the entire data structure whereas the binary search won't.

4. What is Time complexity and space complexity of Binary Search in Best, Average, worst cases.

Ans. Time Complexity :

1. Best Case : O(1). If the calculated mid value is equal to our desired value then there is no need of further calculations. We can simply exit from that loop.

2. Average Case : O(log n). If our desired value is near to mid value but we need to further divide the array so here the divide is compulsory. So it is O(log n).

3. Worst case : O(log n). Even though it checks with the lower bound value or upper bound value it still needs to divide so even though there are more number of divisions of sub arrays but it is still O(log n).

Space Complexity : Even though we are using variables for storing lower bound and upper bound as they are constant variables the space complexity is O(1) only.

5. How to derive the Time Complexity of Binary Search and what is the method we need to follow.

Ans. If the array is just divided into half and the half sub array is traversed or used for the final search fully we can derive the time complexity as O(n/2). But for every iteration it reduces the half of the array which means O(n/2). So using logarithms if something reduces to half every time means O(n) → O(n.2) → O(n/4) → O(n/8) → O(n/16) and so on it goes. So the complexity is O(log n).

Method to Follow : The popular recurrence relation using substitution method is the method we need to follow to solve the recurrence relation. This method is mainly known in MFCS/DMGT.

6. What is Ternary Search and how it is different to Binary Search and how it is similar to Binary Search.

Ans. Ternary Search : It is a searching technique where the input data structure must be sorted and in this the data structure will be divided into 3 sub arrays every time which means at initial we define lower bound and upper bound after one iteration if the desired value is not found then definitely the array is divided into 3 sub arrays.

How this is different to Binary Search :

1. In terms of Complexity : Even though the array is divided into 3 halves but the searching is overhead because of very comparisons. That's why its time complexity is O(log n_3).

2. In terms of division : As we have discussed just now the input array is divided into 3 sub arrays every time where as the binary search is only divided into 2 sub arrays.

3. In terms of Middle values : In this we calculate 2 middle values whereas in binary search we calculate only one middle value.

How this is similar to Ternary Search :

1. In terms of input : In ternary search also we need input data structured to be sorted just like in binary search.

2. In terms of Searching : Just like binary search it also checks the mid value and just uses the 1 sub array and leaves the other 2 sub arrays.

7. Why Binary Search is not called Divide and Conquer and why it is called decrease and conquer method.

Ans. Merge Sort is mainly based on the principle of **divide and conquer** where it divides the array into halves and then sorts the array and then merges the array the main point here is it **utilizes the every sub array** and then merges whereas the **binary search won't need every sub array** it just need one sub array and there is no need of other sub array. That's why it is called decrease and conquer method.

8. When array contains duplicate elements which searching technique by default finds the first occurrence of element ? Why ?

Ans. It is **Linear Search** and only when we start iterating the array from the start because as we are iterating the array from start by checking one by one value and not skipping any value so linear search is the one.

9. If there are duplicate elements in sorted array and by default binary search does not guarantee first occurrence why?

Ans. When there are duplicate elements in the sorted array there is no sure that middle value will be always the fist occurrence of the duplicate values. Because binary search will start checking by the middle value if the middle value is the desired value and that is the 4th occurrence of the value then binary search will return its position.

Ex : [2, 3, 4, 5, 5, 5, 5, 5, 5, 6, 7, 8] , target = 5.

Here we will get the position as 5 because the element is 5 as position 5. and that's not first occurrence.

10. Which searching technique is most suitable to search data in linked list?

Ans. It is **Linear Search** because it just goes by checking one by one value from the start as the linked list have head on any type of linked list. When we assign start as head value it just goes to next value by changing the address of next value.

11. Though data in Linked List is sorted we cannot perform binary search and ternary search on linked list. Why ?

Ans. The main issue with binary search or ternary search is it calculates middle values and then checks with our desired value. But in the linked list the values are stored in **non contigous memory locations** so we cannot access that middle value as we don't have it's address on our hands right? So if we start from the beginning only it is possible to get the middle value that too we cannot be sure which is the middle element until we traverse the entire linked list because if we reach the end of the linked list and then we can check where could be the middle value by making half of the size.

12. what is sentinel search? How it is different to linear search ?

13. what are the drawbacks of sentinel search ?

-By Komesh(23A81A0509)