Commonly Asked Algorithm Interview Questions | Set 1

What is an algorithm?
Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output. (Source: Introduction to Algorithms 3rd Edition by CLRS)

What is time complexity of Binary Search?
Time complexity of binary search is O(Logn). See Binary Search for more details.

Can Binary Search be used for linked lists?
Since random access is not allowed in linked list, we cannot reach the middle element in O(1) time. Therefore Binary Search is not possible for linked lists. There are other ways though, refer Skip List for example.

How to find if two given rectangles overlap?
Two rectangles do not overlap if one of the following conditions is true.
1) One rectangle is above top edge of other rectangle.
2) One rectangle is on left side of left edge of other rectangle.
See Find if two rectangles overlap for more details.

How to find angle between hour and minute hands at a given time?
The idea is to take a reference point as 12. Find the angle moved by hour and minute hands, subtract the two angles to find the angle between them. See angle between hour hand and minute hand for more details

When does the worst case of QuickSort occur?
In quickSort, we select a pivot element, then partition the given array around the pivot element by placing pivot element at its correct position in sorted array.
The worst case of quickSort occurs when one part after partition contains all elements and other part is empty. For example, if the input array is sorted and if last or first element is chosen as a pivot, then the worst occurs. See https://www.geeksforgeeks.org/quick-sort/ for more details.

A sorted array is rotated at some unknown point, how to efficiently search an element in it.
A simple approach is linear search, but we can search in O(Logn) time using Binary Search. See Search an element in a sorted and pivoted array for more details.
Other variations of this problem like find the minimum element or maximum element in a sorted and rotated array.

Given a big string of characters, how to efficiently find the first unique character in it?
The efficient solution is to use character as an index in a count array. Traverse the given string and store index of first occurrence of every character, also store count of occurrences. Then traverse the count array and find the smallest index with count as 1. See find the first unique character for more details.

How to count inversions in a sorted array?
Two elements arr[i] and arr[j] in an array arr[] form an inversion if a[i] > a[j] and i < j. How to count all inversions in an unsorted array. See Count Inversions in an array for all approaches.

Given a big array, how to efficiently find k'th largest element in it?
There can be many solutions for this. The best solution is to use min heap. We Build a Min Heap MH of the first k elements. For each element, after the kth element (arr[k] to arr[n-1]), compare it with root of MH, if the element is greater than the root then make it root and call heapify for MH, Else ignore it. Finally, MH has k largest elements and root of the MH is the kth largest element. See k largest(or smallest) elements for more details.

Given an array of size n with range of numbers from 1 to n+1. The array doesn't contain any duplicate, one number is missing, find the missing number.
There can be many ways to solve it. The best among is to use XOR. See Find the missing number for details. There are many variations of this problem like find the two repeating numbers, find a missing and a repeating number, etc.

How to write an efficient method to calculate x raise to the power n?
The idea is to use divide an conquer here to do it in O(Logn) time. See Write a C program to calculate pow(x,n) for more details.

Given an input string and a dictionary of words, find out if the input string can be segmented into a space-separated sequence of dictionary words.
The idea is to use Dynamic Programming. See Word Break Problem for more details.

Given a row of n coins of values v1 . . . vn, where n is even. We play a game against an opponent by alternating turns. In each turn, a player selects either the first or last coin from the row, removes it from the row permanently, and receives the value of the coin. Determine the maximum possible amount of money we can definitely win if we move first.
This is also a Dynamic Programming Question. See Optimal Strategy for a Game for more details.

You are given an array of sorted words in an arbitrary language, you need to find order (or precedence of characters) in the language. For example if the given arrays is {"baa", "abcd", "abca", "cab", "cad"}, then order of characters is 'b', 'd', 'a', 'c'. Note that words are sorted and in the given language "baa" comes before "abcd", therefore 'b' is before 'a' in output. Similarly we can find other orders.
This can be solved using two steps: First create a graph by processing given set of words, then do topological sorting of the created graph, See this for more details.

You may also like

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Attention reader! Don't stop learning now. Get hold of all the important DSA concepts with the **DSA Self Paced Course** at a student-friendly price and become industry ready. To complete your preparation from learning a language to DS Algo and many more, please refer **Complete Interview Preparation Course.**

In case you wish to attend **live classes** with experts, please refer **DSA Live Classes for Working Professionals** and **Competitive Programming Live for Students**.

My Personal Notes *arrow_drop_up*