University of Nevada, Las Vegas Computer Science 477/677 Fall 2022

Welcome to my CS477/677 class.

What is the goal of this class? For you to learn some basic data structures and algorithms that will help you in the future. I will assume that most of you are planning a career that involves computers, and will almost surely involve programming. Here is a list of possible interview questions. I got most of them from lists of questions that were actually used used for hiring at Google or Facebook.

1. Given an array of integers of length n and a value m, determine whether there are any two integers in that array whose sum is m.

   You can easily prove that the time complexity of this problem is $\Omega(n)$, since you might have to look at every entry. I can solve it in $O(n \log n)$ time. Can it be done faster? I don't know.

2. Given a binary tree, swap the left and right children of every node.

   This can be done in $O(n)$ time.

3. Check whether two binary trees are identical.

4. Given a dictionary and a word w, determine whether w is the concatenation of words in the dictionary.

5. Find all palindromic substrings of a given string.

6. Given an array of integers, including both positive and negative integers, find the maximum sum of any contiguous subarray.

   I know of four algorithms for this problem.
   The dumb one takes $O(n^3)$ time: find the sum of each contiguous subarray, then pick the maximum.
   With a little bit of thinking, you can reduce the time to $O(n^2)$.
   There is also an $O(n \log n)$ time algorithm.
   Finally, there is a linear time (that is, $O(n)$-time) dynamic programming algorithm.

7. We define a string w to be a *numeral* if it satisfies the following conditions.

   (a) The string is not empty.

   (b) Each symbol of the string is either a minus sign, a digit in the range $0 \ldots 9$, or a decimal point.

   (c) There can be at most one decimal point.

   (d) If there is a minus sign, it must be the first symbol.

   (e) If there is a decimal point, there must be a least one digit to the left and at least one digit to the right of the decimal point.

   (f) The first digit to the left of the decimal point cannot be 0 unless it is the only digit to the left of the decimal point.

   (g) If there is no decimal point, the first digit cannot be 0 unless it is the only digit.

   (h) If there is a decimal point, the last digit cannot be 0.

   (i) The string "-0" is not permitted.

   Find an algorithm which reads a string w and decides whether w is a numeral.

8. Print all balanced strings of left and right parentheses. For example, "()(())" and "((()()))" are balanced, but "())(()" is not. Since the number of such strings of length n is $\binom{2n}{n}$, which is an exponential function of n, there is no polynomial time algorithm for this problem.

9. Find the minimum spanning tree of a connected undirected graph with weighted edges.

10. *Least Recently Used*, LRU, is a common caching strategy. If the cache is full and a new item is requested, the item in the cache which was least recently requested is evicted. Design a data structure which implements LRU.

11. Given an array A of length n, where n could be in the millions, and given a value x, there could be many copies of x is the array. Assuming A is sorted, find the first and last indices of A for which the entry is x.

12. You are given an array (list) of interval pairs where each interval has a start and an end time. The input list is sorted by start time. You are required to merge overlapping intervals and return the output list.

    Example: If the input list is (0,5), (3,8), (10,12), the output list should be (0,8), (10,12).

13. Given an integer array, move all elements equal to 0 to the left while maintaining the order of the other elements.

    Example: If the input array is 0,6,4,0,9,1,0,2,0 the output is 0,0,0,0,6,4,9,1,2.

14. Suppose non-negative integers are represented as linked lists where each node holds a digit. (This allows us to represent an integer of enormous size.) Given linked lists representing positive integers n and m, output the list which represents $n + m$.

    Example: if the two input lists are $9 \to 5 \to 8 \to 0 \to 5$ and $6 \to 1 \to 7 \to 8$, the output is $1 \to 0 \to 1 \to 9 \to 8 \to 3$

15. Merge two sorted linked lists into a sorted linked list.

16. Convert a binary tree into a doubly linked list. The order of the items in the list must be the same as the in-order of the tree, and each node of the list must have a forward and a back pointer.

17. Level order traversal of a binary tree. Given a binary tree, output a list whose entries are the same as those of the binary tree in level order.

18. Given a binary tree where each node has an integer weight, and given an integer S, find all root-to-leaf paths the sum of whose weights is S.

19. Reverse the words in a sentence given as a string. If the input is "dog bites man" the output is "man bites dog"

20. Find the maximum single sell profit. Given a list of daily prices of a given stock, find the dates you should have bought and then sold to maximize your profit.

21. Write a function which computes an integral power of a real number. If the input is the ordered pair (x, n) where x is a double and n is an integer, power(x,n) returns the value of x^n. Your function must work for negative integers and zero as well as positive, and for all reals regardless of size or sign. But if $x = 0.0$ and n is negative your function should give an error message.

22. Serialize a binary tree. That means, store a binary tree as an array, in such a way that the binary tree can be recovered from the array. You may not assume that the tree is partially complete.

23. Search a rotated sorted array. Write an algorithm which searches for a specific value in an array which was first sorted and then rotated by an arbitrary number. Example: The input array could be 8, 19, 23, 37, 2, 3, 6

24. We all know how to delete a node in a linked list in $O(1)$ time given a pointer to its predecessor node. How would you delete a node in a linked list in $O(1)$ time if you are only given a pointer to that node?

25. Let $F(n)$ be the n^{th} Fibonacci number. How would you compute $F(n)\%m$ quickly if n and m are huge, say in the billions?