Indian Statistical Institute
Computer Science 3
Third Year Students
10Mar2022

Total Marks: 70

ANSWER ALL Questions

Question 1 [Total Marks: 3x5 = 15]

Give very brief answers (each answer preferably not exceeding 5-7 lines)

a. What is the order of adding an element in any place in an array (assume that array has n elements)? Justify your answer. (By order we mean complexity in big O notation.)

b. Arrange the following in the correct increasing (fastest to the slowest algorithm) order:
$O(n), O(2^n), O(n^3), O(n \log n), O(n^2), O(\log n), O(n!)$ (explanation not needed)

c. What is the main reason for recursive algorithms for functions such as a function generating Fibonacci numbers to be slow? Explain with an example.

d. In insertion sort of n elements, suppose we have an array where first n-1 elements are already sorted and only the last element has to be inserted in the right place. Since the first n-1 element are already sorted, we can do a binary search to find the place of insertion for the last element. Since binary search will takes O(log(n-1)), the time taken to sort such an array should be O(log n). Explain why this reasoning is wrong and describe the correct complexity for this last step.

e. In developing an algorithm there is usually a brute force approach that is guaranteed to work but has significantly high time complexity. One way to reduce the time complexity is to store and reuse previously computed values or to use precomputation that can be used more efficiently. We discussed several examples of these in the class. Briefly describe ONE such example explaining in words (without calculation that is) how it reduces time complexity in your example.

Question 2 [Total Marks: 9+6=15]

2a.) Write an algorithm for a function that will accept a sorted integer array and its length as inputs (the array may have duplicate elements). The function should compact it and then return the new length of the array without duplication. For example, if the input array contains: 30, 34, 39, 47, 47, 59, 69, 69, 70 when the function returns, the contents of the first seven elements of the array should be: 30, 34, 39, 47, 59, 69, 70 with a length of 7 returned.

2b.) What is the order of the algorithm in the big O notation?

Question 3 [Total Marks: 5+1+4=10]

We discussed the stock span problem in the class. Given an array of prices of a single stock for N number of days, stock span for each day is defined as a number of consecutive days prior to the current day when the price of a stock was less than or equal to the price at current day. The input for this problem is an array containing the price for the stock price for a given number of days. The output is the span array. An efficient algorithm uses a stack to store the index of the original price array and is summarised below.

1. Initialize span of day 1 (i=0) as 1 and put 0 on the stack.
2. For i=1 to N-1, do following
3. While price[stack.top()] < price[i]   AND !stack.isEmpty(), stack.pop()
4. If price[stack.top()] > price[i] , set span[i] = (i – stack.top())
5. Push current day index i on to stack

Suppose the prices are as follows: price[]={105,65,75,70,85,90},

Q3a. Write the content of the complete stack on day 4 (i=3) and day 5(i=4).
[For your reference, note that on day 2 (i=1) stack =[0,1] with 1 on top]

Q3b. Write the span[] corresponding to the above price array. No explanation needed.

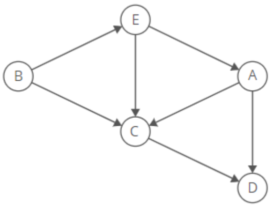
Q3c. What is the time complexity of the above algorithm in big O notation? Justify your answer.

Question 4 [Total Marks: 6+2+3=11]

You are given a linearly linked list of N nodes. But you suspect it has cycles in it. Develop an algorithm (a pseudo code is enough) to detect this cycle. What would you expect to be output of such an algorithm? Your answer should be directly implementable in a C code, though you do not have to write the C code. In terms of N, what is the complexity of your algorithm in big O notation?

Question 5 [Total Marks: 7+4x2+4=19]

We discussed in the class a queue based algorithm for topological sort of a Directed Acyclic Graph. Use that algorithm to answer the following questions.



a. With the help of diagrams, show ALL the steps involved in producing a topological sort of the above diagram. Your description must indicate the output array and the queue used as well as the processes involved. Feel free to use diagrams to denote the intermediate states of the graph above.

   Suppose an arbitrary DAG has V vertices and E edges. Answer the following questions for such a graph. Remember that The number of edges coming into a vertex in a directed graph is the in-degree of that vertex.

b. What is the order of the computation to calculate the initial in-degree of each vertex?

c. What is the total order of computation to re-calculate the in-degrees as the vertices of in-degree are removed?

d. What the total order of computation to queue and de-queue the vertices.

e. Combining the result from step b.) c.) d.) determine the order of time complexity of queue based topological sort algorithm.

f. Provide an argument, preferably using a loop invariant, that the queue based topological sort algorithm is correct.