Original

Dr Nitin Gupta
Do CSE
227
9/12/22

National Institute of Technology, Hamirpur
Department of Computer Science and Engineering
B.Tech. ECE (Section-I)(End-Term Examination)
CS-201 Data Structure

Time: 3 hrs **Max. Marks 50**

Instructor: Nitin Gupta

For all problems, expand every function that your are calling. Further, if you include any new algorithm in your solution, please also include a brief English description of what the algorithm does.

1. (4 × 5) Answer the following briefly

   (a) What is the need for using circular array to implement queues?

   (b) How a priority queue can be implemented with help of Heap?

   (c) Write an algorithm to concatenate two linked lists?

   (d) Binary search is always faster than a linear search. True or False? Support your answer with all valid points.

2. (10 Marks) Write algorithm to solve an infix expression using a stack. Show the step by step execution of your algorithm on the following expression:

$$(((6 - (2 + 3)) \times (3 + (8/2))) \uparrow 2) + 3$$

3. (10 Marks) Recall that in Huffman coding problem, we are given a set of n characters with some frequencies. We are required to represent each character by unique codeword using 0's and 1's, such that no codeword is prefix of another. The goal is to find out such codeword to achieve maximum compression. Write an algorithm and find out that how many bits will be required to store 'abbccddeee' using Huffman coding. To obtain the unique Huffman tree, we will use FCFS rule where the characters that is already in the Min. priority queue will be extracted first than the character which entered later in the queue. Also characters having same frequency will be extracted first which appears first in the above sentence. e.g. if 'a' and 'e' have same frequency then 'a' will be extracted first.

[*P.T.O.*]

228

4. Write an algorithm to find the Topological sort of a graph. Please note that you have to expand every function that you are calling in your algorithm. Apply your algorithm on the Fig. 1.

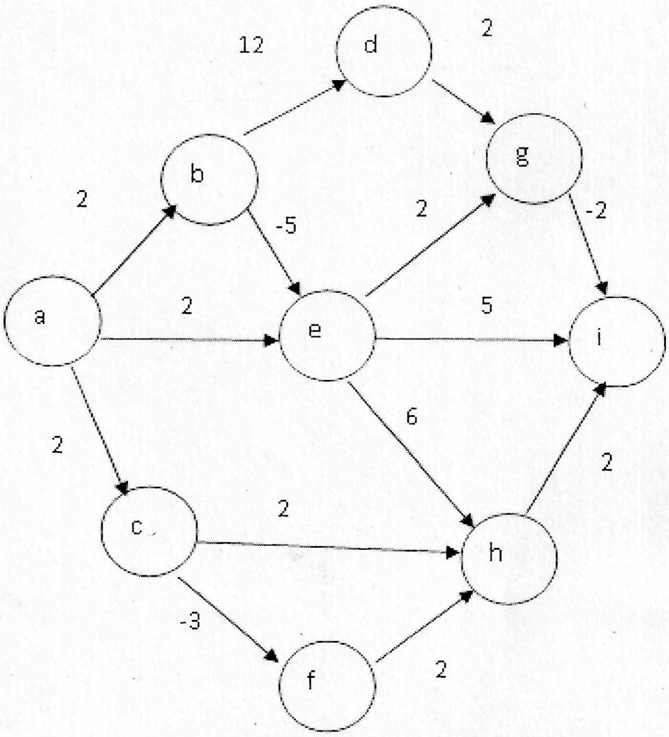

Figure 1

☆ ☆ ☆ ☆