Algorithms-1 - CS21003
(Class Test II)
Date: 21 – November – 2020

Maximum marks: 42 Duration: 1 hour

File naming convention: e.g., 18CS3004_G3_CT2.pdf (or any other extension).

In case of multiple files, use _1, _2 etc at the end.

Submission is via Moodle only. Email submissions will NOT be accepted. Please manage your time well keeping in mind that Internet and power disruptions are a new normal!

No clarifications from the TAs today. You can make any assumption as long as it is rational and you clearly state the same while solving the problem.

Plagiarism, in any form (including Internet source) will be severely penalized.

Question 1

Give the visited node order for each type of graph search, starting with s, given the following adjacency lists and the accompanying figure (Figure 1):

$adj(s) = [a; c; d]$,
$adj(a) = []$,
$adj(c) = [e; b]$,
$adj(b) = [d]$,
$adj(d) = [c]$,
$adj(e) = [s]$.

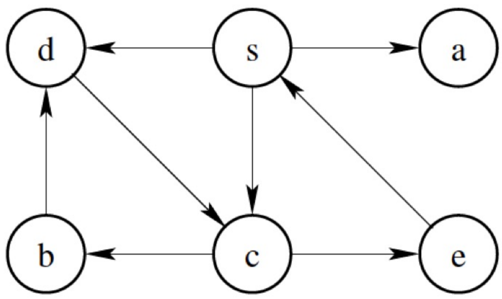


Figure 1: Example graph.

Graph search techniques: (a) BFS, (b) DFS.

[5 marks]

Question 2

Answer **true** or **false** with **short justifications**.

(a) While running DFS on a directed graph, if from vertex u we visit a finished vertex v, then the edge $(u; v)$ is a cross-edge.

(b) Let P be a shortest path from some vertex s to some other vertex t in a directed graph. If the weight of each edge in the graph is increased by one, P will still be a shortest path from s to t.

(c) Dynamic programming is more closely related to BFS than it is to DFS.

(d) A depth-first search of a directed graph always produces the same number of tree edges (i.e., independent of the order in which the vertices are provided and independent of the order of the adjacency lists).

(e) Suppose we do a DFS on a directed graph G. If we remove all of the back edges found, the resulting graph is now acyclic.

[10 marks]

Question 3

You are given a list of all scheduled daily flights in India, giving departure airports, departure times, destination airports, and arrival times. We want an algorithm to compute travel times between airports, including waiting times between connections. Assume that if one flight arrives at time t and another departs at time $t' \geq t$, travelers can make the connection. Further assume that at a given airport, no two events (arrivals or departures) occur at the same time, and that there are at most 100 events at any airport during a given day. All times are given in GMT and India has a single time zone. We can construct a weighted graph so that given a departure airport, a departure time T, and a destination airport, we can efficiently determine the earliest time T' that a traveler can be guaranteed (according to the schedules!) of arriving at her destination on that day (ignore overnight flights and overnight airport stays).

We create a vertex for every flight departure and every flight arrival. That is, each vertex corresponds to a particular time (arrival or departure of a flight) at a particular airport. We create an edge for every flight. The weight of these edges is the flight time. We also create an edge from every event (departure or arrival) at an airport to the next event at the same airport. The weight of these is the time between the events.

- Give an upper bound on the number of edges and vertices in your graph if there are n airports in India and m daily flights. Justify your bound.
- Given that you have constructed the graph, how will you compute the shortest travel times?

[4 + 2 = 6 marks]

Question 4

Answer **true** or **false** with **short justifications**.

(a) There is a heap T storing seven distinct elements such that a *preorder* traversal of T yields the elements of T in sorted order. If your answer is **true**, draw the heap. If your answer is **false**, give a brief justification.

(b) The same problem as above but this time with an *inorder* traversal.

(c) We discussed the median of medians method to select a pivot, that provides a worst-case linear-time algorithm for the k^{th} order statistics. If instead of a group of 5 elements, we take a group of 3 elements, we still get a worst-case linear time algorithm. You may justify your answer using the final recurrence relation, do not need to show its derivation.

(d) For priority queue implementation, we need three main operations: deleteMax, findMax and insert. There is an algorithm that can perform each of these operations in $O(1)$. If your answer is **true**, provide the algorithm. Otherwise, explain briefly why this is not possible.

[8 marks]

Question 5

Suppose you have a book shelf, that can only be accessed from left to right. Suppose you keep n books in the shelf. For each book b_i, you have a probability $p_i > 0$ with which you access, and it has a width w_i. Suppose, the books are placed in an order $b_1, b_2, \ldots, b_n$. To access the book b_i, you need an access time proportional to $\sum_{j=1}^{i} w_j$. Thus, you can define the expected access time for the books as

$$\sum_{i=1}^{n} p_i \sum_{j=1}^{i} w_j$$

You need to design a greedy algorithm to minimize the expected access time. Below, you are given two greedy algorithms.

(a) Order the books from smallest to largest (as per width) on the shelf. That is, $w_i < w_j$ implies that $i < j$.

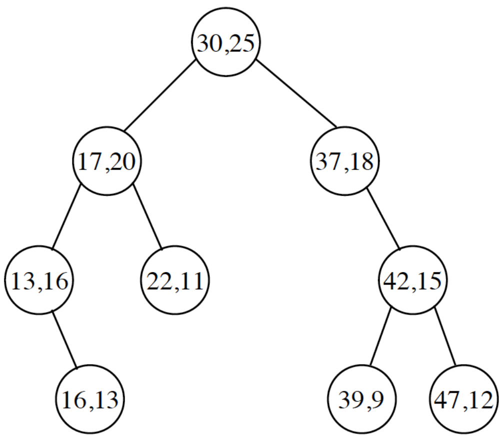
(b) Order the books from most likely to be accessed to least likely to be accessed. That is, $p_i < p_j$ implies that $i > j$.

For each of the greedy algorithms, if the algorithm is correct, mention that. If it is incorrect, provide a counter-example. If both the algorithms are incorrect, you need to provide counter-examples for both, and also mention the correct greedy algorithm. No proofs are required.

[7 marks]

Question 6

Consider the following diagram. Each node stores a pair (x, y), with x denoting a value and y denoting its priority. See that while the values follow binary search tree property, the priority scores follow a heap property (not the heap structure).



- Briefly explain how you can insert a new item (a, b) into this structure so as to follow the above mentioned restrictions.
- Show the modified structure after inserting $(33, 28)$.

[6 marks]