Type: DES

Q1. Open Shortest Path First (OSPF) is a routing protocol used in IP networks. It implements the Dijkstra's algorithm to find the shortest path from a source router to a destination router. A router is a networking device used to interconnect computer networks by forwarding data packets from the source to the destination. A packet is typically forwarded from one router to another router through the networks that constitute an internetwork (e.g. the Internet) until it reaches its destination node. A router maintains a routing table with the IP address of its neighbouring routers and the number of hops to reach the neighbouring routers.

For example: Assume the source router to be at Manipal and the destination router for an email packet to be at Amsterdam. When the packet reaches, an intermediate router at Bangalore, it searches its routing table to forward the packet via the shortest path to reach Amsterdam (either through a single hop or multiple hops). Which data structure, in your opinion, would be used for the storage of the routing table? Justify your choice. (5)

Q2. With a suitable example discuss the drawback of a binary tree and the need for an AVL tree. Perform insertion of the following nodes in an AVL tree and balance it: (3)

Q3. A discrete event simulator for generation of automobile traffic is under construction. It is a simulator for creation of real-world systems as a series of events which occur according to a schedule. The following is the set of events have been considered to construct the simulator:

- A traffic light which turns red, green or yellow after a specified time lapse.

- Parking lots and garages have been simulated to toss out a new car on the road at random intervals of time (more during the rush hours of office traffic).

- Each moving car is associated with a time at which it has to reach the next intersection along the road.

In your opinion, what would be the most ideal data structure to create this series of events in the form of a schedule? Justify your answer. (3)

Q4. Is the following statement true? Justify your answer.

$$If\ f(n) = O(g(n)\ )and\ g(n) = O\big(h(n)\big), then\ h(n) = \Omega(f(n))\quad (2)$$

Q5. Write the algorithm/ pseudo code for Insertion sort. Compare and contrast it with Bubble sort. (3)

Q6. The field of Bioinformatics uses searching algorithms to look for structural patterns in protein DNA. For example in the structure 'ATGACGATAAAGGCACGGC' the pattern 'AAA' has to be searched for. Could you suggest how this can be done using the Horspool algorithm? (3)

Q7. The following is a news snipped from the InfoSecurity magazine. According to the article, what is the main strategy for hacking the passwords of the GitHub users? Explain by using an example.
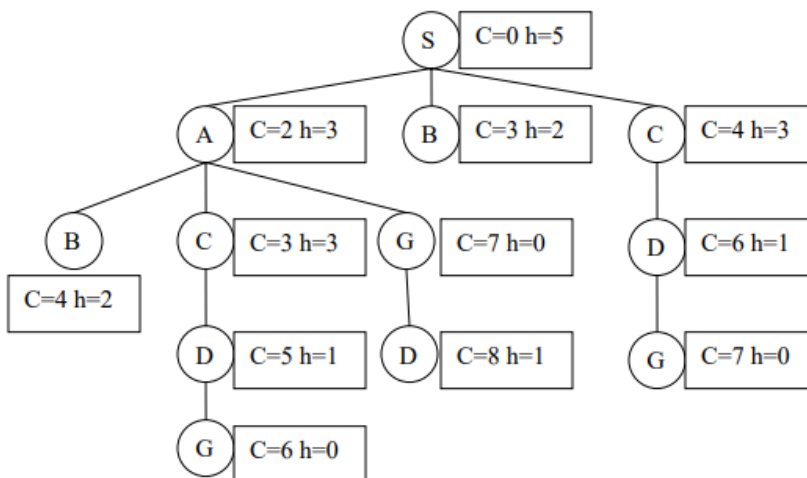
GitHub resets passwords after mass-scale Brute force attack (22nd November 2013)

Brute-forcing, where perpetrators use automated software to methodically attempt to access online accounts using commonly used or easily guessed passwords (i.e., the account-holder's name plus "123"), is not uncommon. But the GitHub attack is notable for its scale: it was carried out from almost 40,000 different IP addresses, which were used to slowly test for weak passwords or those passwords used on multiple sites (always a common foible for end users).

As a result, the company is telling users that it's now requiring them to be more responsible in their password choices. (3)

Q8. How is the problem of collision/ spurious hits resolved in the Rabin Karp algorithm used for text processing? (3)

Q9. Using the following search tree, perform the searches indicated below from S to G. Each node shows both the total path cost to the node and heuristic value for the corresponding node. (7)

For each of the searches below, write the sequence of nodes expanded by the searches.
Specify a node by writing the name of the state and the length of the path (C above), e.g. S0, B3 etc.
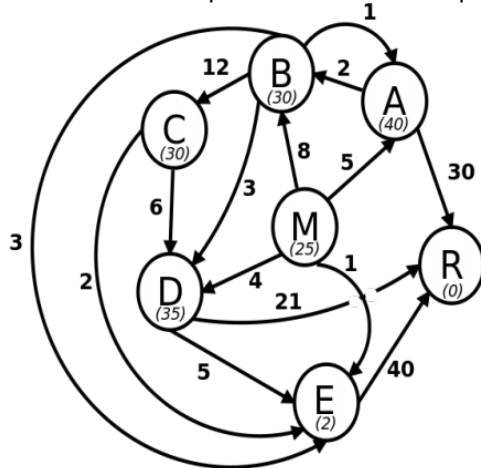Assume that the children nodes are expanded in alphabetical order.

a) Depth First Search (with no memory of Visited Nodes)
b) Breadth First Search (with memory of Visited Nodes)
c) Dijsktra Search Method
d) A* Method

Q10. As you get close to graduating MIT, you decide to do some career planning. You create a graph of your options where the start node is **M = MIT** and your goal node is **R = Retire**, with a bunch of options in between. Your graph includes edge distances that represent, roughly, the "cost of transition" between these careers. You also have heuristic (in parenthesis) node-to-goal distances which represent your preconceptions about how many more years you have to work until you retire.

For example, you think it will take 25 years to go from MIT (M) to retirement (R), 30 years from Grad School (B), but only 2 years from Entrepreneur (E).

A = Wall Street | B = Grad School | C = Professor | D = Government | E = Entrepreneur
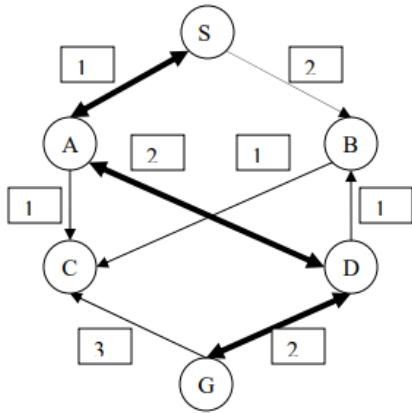

a) Assume you want to retire after doing the least number of different jobs. Of all the basic search algorithms you learned about which one should you apply to the graph in order to find a path, **with the least search effort**, that has the minimum number of nodes from M to R? Draw the tree and give the path. (3)

b) Now you are interested in finding a path and the associated distance. Due to Space Constraints you can store total cost information of only two nodes. As before, you are looking for a path from M to R. Draw the tree and give the path.
(3)

Q11. Illustrate the supremacy of A* Search Technique over Greedy Search Techniques. Explain, Admissibility and Consistency of Heuristics. (5)

Q12. Construct the search tree for the graph above, indicate the path length to each node. The numbers shown above are link lengths. Pay careful attention to the arrows; some are bi-directional (shown thick) while some are uni-directional. (7)

Choose the most efficient search method that meets the criteria indicated below. Explain your choice.

a) You are given a state graph with link costs. The running time of the algorithm should be a function of the number of states in the graph and the algorithm should guarantee that the path with shortest path cost is found.

b) You are given a state graph with link costs and consistent heuristic values on the states. The running time of the algorithm should be a function of the number of states in the graph and the algorithm should guarantee that the path with shortest path cost is found.