CS2040S Final Assessment

MCQ

Q1: Given a BST (which is not balanced) of size **N**, to convert it to a balanced BST (according to AVL property) requires at least:

a. O(1) time
b. O(log N) time
c. O(N) time
d. O(Nlog N) time

Q2: You are given a binary max heap, which uses an array implementation (as covered in lecture). N is known, and is guaranteed to be odd. If we want to find the median value of the heap, we can do so in:

a. O(1) time
b. O(log N) time
c. O(N) time
d. O(N log N) time

Q3: Given an undirected graph, we want to find out if this graph is a tree. Initially, the only information you have is that this graph is an undirected graph. No additional information is provided. Which of the following combinations of information will be sufficient to deduce, **for all possible graphs**, if the graph is a tree?

i. Number of vertices in the graph
ii. Number of edges in the graph
iii. In/out degree of each vertex
iv. All edges, given in the form of an adjacency list

a. (i) only
b. (i) and (ii)
c. (i), (ii) and (iii)
d. (i), (ii), (iii), and (iv)

Q4: You are given an initial list of integers, which you should store in a DS of your choice. Afterwards, your DS should be able to support the **removeMin()** and **removeMax()** operations, which removes and returns the smallest/largest element respectively from the DS. It is guaranteed that **no other operations will be performed on the DS**. In order to **minimise the total time complexity** of the removeMin() and removeMax() operations, which DS would be the most ideal?

a. Hash Table
b. Sorted doubly linked list
c. Minimum binary heap
d. AVL tree

Q5: The following 5 strings are inserted into an initially empty Trie:

bear
bold
cold
cord
dear

How many nodes (including the root) will be present in the Trie after all strings are inserted?

a. 9
b. 10
c. 18
d. 21

Q6: You are asked to draw a directed graph with 5 vertices and 7 directed edges. Additionally, the graph must contain the largest possible number of valid topological sorts. How many valid topological sorts are present in this graph?

a. 1
b. 6
c. 24
d. 120

Q7: 8 elements are added one at a time into an initially empty AVL tree. What is the maximum number of rebalancing operations that occurred across all insertions?

a. 3
b. 4
c. 5
d. 6

Q8: Given a directed graph with V vertices, what is the maximum number of directed edges that can be present in the graph without forming any cycles?

a. V(V-1)
b. V(V-1)/2
c. V(V+1)
d. V(V+1)/2

Q9: You want to find out if an **undirected, disconnected graph** is reverse-bipartite (not an actual term). For a graph to be reverse-bipartite, vertices can be assigned to one of two sets, and edges that appear in the graph can only be between two vertices in the same set (so these vertices should not be in two different sets). Your program **only needs to return true/false**. This can be done in:

a. O(1) time
b. O(V) time
c. O(V + E) time
d. $O(V^2)$ time

Q10: You are given an undirected weighted graph G, and the only MST of this graph T. Then, one edge in G (which is also in T) is removed from the graph. You are now required to find an updated MST of G (G is guaranteed to still be connected after the edge is removed). This can be done in :

a. O(V) time
b. O(VlogV) time
c. O(V + E) time
d. O(ElogV) time

Analysis:

1. Given N distinct string keys (N > 1), each of the same length w (w > 2), that is formed from an alphabet set of size R (R > 1), storing all N keys in a Trie cannot take less then O(wNR) space.

2. There is no non-empty binary max heap containing integer keys which also satisfies BST property.

3. In a SCC (strongly connected component), removing any one edge in the SCC will destroy the SCC (meaning not all vertices will be able to visit all other vertices in the SCC after that edge is removed).

4. Floyd Warshall is the most efficient algorithm for solving APSP for all kinds of graphs.

Structured Questions

1. Given an AVL storing N distinct integer keys, give an algorithm to delete all keys that fall within the range from i to j (i <= j, i and j inclusive) in worst case time <= O(KlogN), where K is the number of keys to be deleted. You cannot use Java library TreeSet or TreeMap in your algorithm. You can only use the AVL operations discussed in lecture. If you need to modify any of the operations, describe your modification.

2. Given an unweighted graph (with at least 1 edge) stored in an adjacency list, give an algorithm that will return 1 if it is an undirected graph, 2 if it is a directed graph and 3 if it is a hybrid graph (a mix of directed and undirected edges). Your algorithm must run in average time <= O(V+E) where V is number of vertices and E is the number of edges in the graph.

3. There is a game where you have N dots numbered from 1 to N, and for some pairs of dots A,B there is an arrow going from A to B (if there is an arrow going from A to B then there will not be an arrow going from B to A).
The objective of the game is to place the pencil at some starting dot and trace the arrows in such a way as to go through as many dots as possible without lifting the pencil (you may traverse some arrows multiple times to achieve this).
Given there is at least 1 starting dot A' which allows to you to trace the arrows in the above stated way and go through all the other dots, model the game as a graph and give the most efficient algorithm in terms of worst case time complexity you can think of to return one such A'.

4. Routers in a network are often susceptible to failure. Dr Ferdaus of Ferulock fame/infamy has developed his own protocol to test if a router in a network is still functioning.

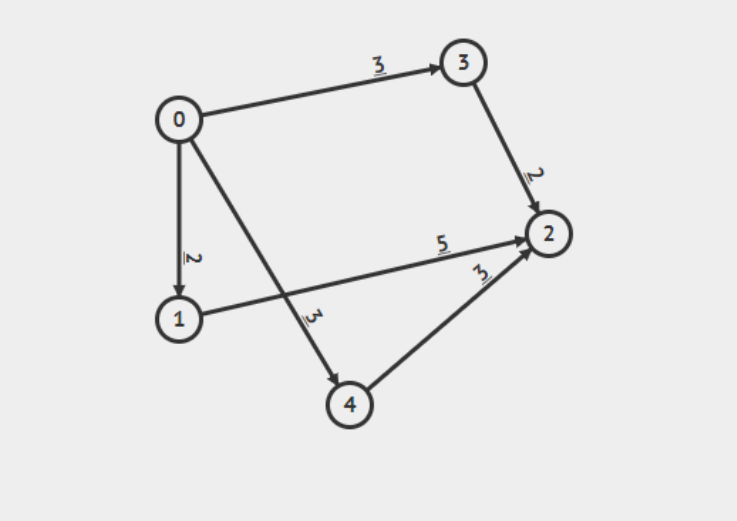
Given a graph representing the network, vertices are the routers (there are V number of them numbered from 0 to V-1) and the routers are connected by E directed edges ($\frac{V(V-1)}{10} \leq E \leq \frac{V(V-1)}{2}$). For any pair of router A and B connected by a directed edge from A to B with integer weight W(A,B) (W(A,B) >= 1), A will send a ping request to B at intervals of W(A,B) seconds (you can assume the time to send the request to B is 0 seconds). If B is alive it will have to send a default message back to A. All the routers in the network is synchronized to a global clock, so they will start counting of their intervals to send pings at the same time.

Now given that the graph as described above is stored in an adjacency list, give the best algorithm in terms of worst case time complexity you can think of to answer the following query (there can be many of such queries):

RequestList(i,K) - Return the 1st K vertices that sends a ping request to vertex i ordered by increasing time of request. Number of vertices sending pings to i <= K <= V. If two different vertices sends a request at the same time, they should be ordered by increasing vertex number.

If required, you can perform pre-processing that takes no more than O(V+E) time. Describe the algorithm for your pre-processing too.

In the example graph given below that represents a network of routers, RequestList(2,4) will return {3,4,3,1} as the first 4 vertices sending pings to vertex 2 in order of increasing time of request. The time of their requests are {2 secs, 3 secs, 4 secs, 5 secs} respectively.



5. Given a graph G that is a DAG with V vertices and E edges (where E = O(V)) stored in an adjacency list, you want to find the weight of the largest edge along the minimax path from a given source vertex A to a given destination vertex B. Give the best algorithm in terms of worst case time complexity you can think of to do it. You may assume there is at least 1 path from A to B.

6. The salesman from tutorial 11 has begun another round of travelling around different cities and peddling his wares. This time he has set aside funds to pay the toll fee for every city he passes when getting from some source city A to some destination city B. Of course he still wants the shortest route to get from A to B as time is of the essence. However he has calculated that he has only enough money to pay the toll fee of at most K cities where 1<= K <= 10, thus he cannot pass through more than K cities when getting from A to B (including A and B themselves).

Given the value K and a graph G with V vertices and E edges, where the vertices are cities and bi-directional edges are roads connecting pairs of cities, and edge weight is the travel time (same in both direction), give the best algorithm you can think of in terms of worst case time complexity to find the cost of the shortest path the salesman should use to get from some city A to some other city B that does not involve more than K cities. If no such path exists output "no valid path from A to B".

7. The global merchant guild has set her sights on being the most represented guild on the planet. In order to do so, the best way is of course to build an office in every country on the planet. However, the cost is too prohibitive and also some countries are at war with neighbouring countries and it would not be profitable for the guild to do business in such countries. After analyzing all the N countries (the countries are numbered from 0 to N-1) and their relationship to each other in terms of distance, politics etc.., the guild has come up the following possible relationships between a pair of country A,B.

A B 1 - Exactly one office will be built in either A or B because they are close to each other and you do not need one office in both A and B.

A B 0 - No office should be built on both A and B because they are at war and having a presence in either country would not be profitable at this time

A B 2 - An office should be built on both A and B because they are far apart but have very strong economic ties and to maximize profit it would be expedient to build an office in both countries.

Given M such unique pairs of countries and their relationships (there is only 1 relationship per country pair, e.g if there is 0 2 1, there will not be 0 2 0 or 0 2 2), determine the minimum number of offices to build to satisfy the M relationships so as to achieve maximum profitability and also which countries they should be built in. If it is impossible to satisfy the M pairs of relationships, simply return -1.

Give the best algorithm in terms of worst case time complexity you can think of to solve the problem.