CS2040 2020/2021 Semester 2 Final Assessment

MCQ

This section has 10 questions and is worth 30 marks. 3 marks per question.

Do all questions in this section.

1. You are given a connected, undirected graph, where all edges have the same weight. You are asked to find a valid MST of this graph, by printing out the edges belonging to this MST. If there are multiple valid MSTs, you only need to output any one of them. The best algorithm to do so runs in worst case

a. O(1) time
b. O(V) time
c. O(V + E) time
d. O(E log V) time

2. Which of the following operations will run in worst case O(log n) time or better? Assume that the following data structures only contain **n** distinct keys.

i. Removing the smallest element in a binary max heap
ii. Removing the smallest element in a BST
iii. Removing the smallest element in an AVL tree

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

3. There are 24 different ways to insert each of the integers 1, 2, 3, 4 one at a time into an initially empty AVL tree. Of these, how many of them will result in at least one rebalancing operation?

a. 8
b. 12
c. 16
d. 20

4. You are given a special undirected graph with **V** vertices (**V** is guaranteed to be a multiple of 4), and every vertex has exactly one edge adjacent to it. What is the minimum number of connected components in this graph?

a. 1
b. 2
c. V/4
d. V/2

5. In any valid binary max heap, we can obtain a list of the elements in sorted order by doing:

a. Preorder traversal
b. Inorder traversal
c. Postorder traversal
d. None of the traversal algorithms work

6. You are asked to draw a simple directed graph with 5 vertices, which form a total of 2 strongly connected components. The maximum number of edges in this graph is:

a. 10
b. 12
c. 14
d. 16

7. You are given an undirected graph G, in a graph data structure of your choosing. Which of these DSes will allow you to check which vertices are reachable from a vertex X in worst case O(V + E) time or better?

i. Adjacency Matrix
ii. Adjacency List
iii. Edge List

The following options are independent (eg. picking (i) and (iii) means that it is possible to achieve this with either an adjacency matrix only, or an edge list only)

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

8. Below is the p array of a UFDS. Union-by-rank and path compression are not used in this UFDS.

Index	0	1	2	3	4	5	6	7
p	2	1	2	1	3	4	4	5

What are the sizes of each set in the UFDS? Eg. an answer of 3,5,5 means that there is 1 set with size 3, and 2 sets with size 5.

a. 1, 1, 2, 2, 2
b. 2, 2, 4
c. 2, 6
d. At least one set in this array is invalid

9. The following is an adjacency list of a directed, unweighted graph G. The value of some fields are unknown, however it is known that the list of neighbours for each vertex is always in ascending order.

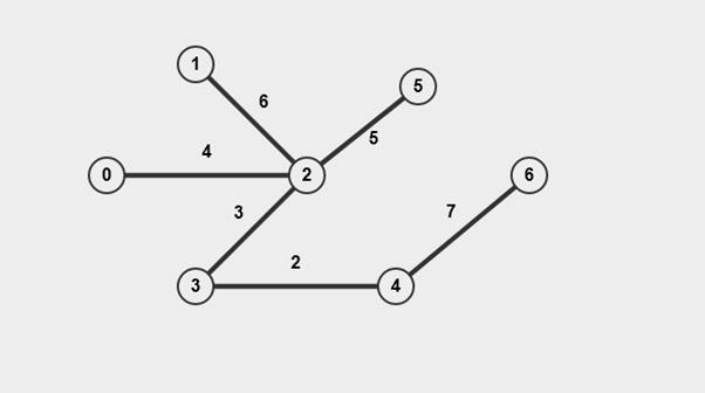
0	2	
1	?	4
2	3	
3	1	
4	?	
5		

Given this information, deduce if the graph may have the following properties. The options are independent (ie. picking (i) and (iii) means that it is possible for the graph to be acyclic (by filling in the unknown cells in a certain manner), or consist of more than one SCC (by filling in the unknown cells with a possibly different set of values from (i))).

i. G may be acyclic.
ii. G may be cyclic.
iii. G may consist of two SCCs or less.

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

10. The following is one possible valid minimum spanning tree of a graph with 7 vertices:



It is known that the original graph consists of 9 edges. Determine the smallest possible total cost of the 3 edges which are not in the MST.

a. 9
b. 10
c. 11
d. 12

Analysis

This section has 4 questions and is worth 12 marks. 3 marks per question.

Please select True or False and then type in your reasons for your answer.

Correct answer (true/false) is worth 1 marks.

Correct explanation is worth 2 marks. Partially correct explanation worth 1 marks.

Do all questions in this section.

11. For fast heap create, we can still get a valid max heap of size N in $O(NlogN)$ time if we go from index N back to index 1 but call shiftUp instead of shiftDown.

12. Given an array of N integer, inserting all of them into a PQ implemented using an AVL tree can achieve the same time complexity as a PQ implemented using a min heap in the worst case.

13. In a weighted undirected graph **G** where the edge weights are all distinct, the minimax path between a given pair of vertices **A** and **B** must only be the path from **A** to **B** in the MST of **G**.

14. Given the keys of a BST in in-order sequence and no other information, we can always reconstruct the exact same BST from the given sequence.

Structured Questions

This section has 6 questions and is worth 58 marks.

Write in **pseudo-code**.

Any algorithm/data structure/data structure operation not taught in CS2040 must be described, there must be no black boxes.

Partial marks will be awarded for correct answers not meeting the time complexity required.

15. [8 marks] Given an AVL tree **A** of size N containing unique integer keys, describe an algorithm to perform the following operation:

    GetAll(left,right): store all keys in **A** where **left** ≤ key ≤ **right** in an arrayList **B** and return **B**

    This operation should take in the worst case $O(logN + K)$ time where K is the number of nodes where left ≤ key of node ≤ right

16. [6 marks] Given a min heap **A** of size N containing integer keys, describe an algorithm to perform the following operation:

    deleteSubTree(val) - delete the entire subtree where the node with key == val is the root. If there is no node with key == val in **A**, nothing is done. After deletion the resultant tree should still be a min heap (meaning that extractMin and insertion operations should still work without any changes).

    This operation should take in the worst case $O(N)$ time.

17. [15 marks] Country X has N states (labeled 0 to $N-1$). When citizens of country X are born, they will register with the state they are born in and be given a unique integer as id.

    During the COVID19 pandemic, zoning by state was implemented, and citizens can only move about within the state they are registered in and cannot cross state lines. However as the pandemic becomes less severe, adjacent states are merged into larger zones where the citizens within can freely move. Ultimately once the pandemic is over, citizens can once again freely move throughout country X.

    In order to track citizen movement during the pandemic, your job as a government employee is to come up with appropriate data structure(s) and algorithms to implement the following operations as efficiently as possible.

    1. **Initialization()** - initialize your DS(es) with a list of size M which contains the id and state number of each citizen. There should be N zones with each state being a zone

    2. **RegisterBirth(v, s)** - register a new birth (using **v** as the unique integer id) with state **s**.

    3. **Merge(s1,s2)** - merge the 2 zones that state **s1** and state **s2** belong to, if they do not already belong to the same zone.

    4. **LegalMovement(v, s)** - check if a citizen with id **v** can legally move about in state **s**. If possible return true else return false. If there is no citizen with id **v**, return false too.

    Please analyze and state the time complexity for the operations you have implemented in terms of N and M.

    Marks distribution:

    Initialization(): 4 marks

    RegisterBirth(v, s): 3 marks

    Merge(s1,s2): 3 marks

    LegalMovement(v, s): 5 marks

18. [7 marks] 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**) . There are M arrows in total.

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 or dots multiple times to achieve this).

Given there is at least 1 starting dot **A'** which allows you to trace the arrows in the above stated way and go through all the other dots, any tracing that will start from **A'** and go through all vertices will stop at some vertex **B'** (note that not all vertices can be a **B'**).

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 possible **B'** that you will stop at after having gone through all vertices from some starting **A'** (**A'** is not given to you).

State the time complexity of your algorithm in terms of N and M.

19. [12 marks] In country Z where the roads are painted red or blue, it will be the inauguration of the new president in a few days. On that day, he will leave his current abode and travel with a motorcade to his new presidential abode.

In order to prevent any unexpected situation from happening enroute to the new abode, the shortest route in terms of distance is preferred by the security team. However, the president is a very superstitious man, and he believes it is auspicious to take only a route that will cross exactly 12 red roads.

To accommodate the president's requirement, the security team has determined there is at least 1 route that will cross exactly 12 red roads. However, they will still want the shortest route that crosses exactly 12 red roads.

Now as a person in charge of decorating both sides of the roads that will be used in the route from the president's old abode to the new abode, you need to determine the total length of decorations to use (in terms of meters). The problem is that the actual route is a secret and is not revealed to you.

You are given a graph **G** that contains the N junctions in country Z as vertices and the M one-directional roads as edges linking junctions/vertices together. For each road you are also given the color of the road and the length in integer meters of the road. You may assume the graph is stored in an adjacency list.

Now given **G**, **s** (the vertex representing the president's old abode) and **s'** (the vertex representing the president's new abode), write the most efficient algorithm you can think of in terms of worst case time complexity to do 1 of the following:

1. Find the total length of decorations to use along all roads on one potential valid shortest route. **6 marks if you get this correct**

2. Find the total length of decorations to use along all roads on all possible valid shortest routes. **12 marks if you get this correct**

If you need to solve the problem by transforming **G,** please describe the transformation before giving the algorithm. Also state the time complexity of your algorithm in terms of N and M

Note that you will need to decorate both sides of the road.

20. [10 marks] Snags are an alien species found on planet Z. They live in subterranean burrows connected by bi-directional tunnels. Each tunnel has a given integer diameter value (same in both direction).

    In order to fit into the tunnels to travel from burrow to burrow, Snags have the magical ability to change their size. To travel from a burrow **s** to another burrow **d**, a Snag will initially expend 0 energy to match the diameter of the first tunnel it takes. Subsequently, it will only need to expend energy to travel through a tunnel of diameter **x** if **x** > **p** (the maximum diameter of tunnels already travelled) or **x** < **p'** (the minimum diameter of tunnels already travelled). Energy expended is **x**−**p** or **p'**−**x** respectively.

    In order to conserve energy, a Snag would like to expend the least total energy to get from some source burrow **s** to some destination burrow **d**.

    Now given N burrows (labelled 1 to N), descriptions of all M bi-directional tunnels (starting burrow, ending burrow and diameter), a source burrow **s** and destination burrow **d**, give the most efficient algorithm in terms of worst case time complexity you can think of to output the least total energy to get from **s** to **d**.

    State the time complexity of your algorithm in terms of N and M.

    You may assume that there is always at least 1 path from any **s** to any **d**.