# 2420 CS2040 Finals

## MCQ: 40 marks, 10 Questions

| |
|---|
| **Q1. [4 marks]** A graph **G(V,E)** is represented as an "adjacency list" with each array entry **Adj[u]** being a hash table containing the vertices *v* for which *(u, v)* ∈ E. If all edge lookups are equally likely, what is the **expected/average time** to determine whether an edge is in the graph? |
| **Q2. [4 marks]** Given a complete undirected graph G(V, E) with V vertices and E edges. What is the most efficient time complexity to traverse the graph breadth-first? |
| **Q3. [4 marks]** Given an adjacency matrix as follows. How many minimum spanning trees do/does the graph have? $$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$ |
| **Q4. [4 marks]** You are given a maximum binary heap **A** of **N** distinct elements, as well as a maximum binary heap **B** of **M** distinct elements.<br>You want to create a new maximum **binary heap C** containing the **N**+**M** distinct elements from **A** and **B**. State the time complexity of the most efficient algorithm to do so, in terms of **N** and/or **M** only.<br><br>All 3 binary heaps are array-based, as described in lectures. |
| **Q5. [4 marks]** You are given a binary search tree **A** of **N** distinct elements, as well as a binary search tree **B** of **M** distinct elements. It is possible that **A** and/or **B** is NOT balanced.<br>You want to create a new **AVL tree C** containing the **N**+**M** distinct elements from **A** and **B**.<br>State the time complexity of the most efficient algorithm to do so, in terms of **N** and/or **M** only. |
| **Q6. [4 marks]** A 0-based UFDS of 8 elements, as implemented in lectures using both union-by-rank and path compression, was initialized as usual into 8 disjoint sets. After some operations, the underlying parents array currently has the following contents:<br>[1, 1, 2, 2, 1, 2, 2, 4]<br>How many of the following statements are correct:<br>i. The UFDS currently has 3 disjoint sets.<br>ii. From initialization to the current state, among all unionSet(x, y) operations performed, exactly 6 of them joined **two different sets** together (i.e. x is a different set from y in that operation)<br>iii. Currently, there exists at least 2 disjoint sets having the same size (number of elements in the set) |

**Q7. [4 marks]** Consider an unweighted DAG with V vertices and E edges. While we know that we can use BFS or a one-pass Bellman-Ford to solve for SSSP in an unweighted DAG, we sometimes need to compute the longest path from a single source instead.
Let's try augmenting algorithms that you have learnt in this course.
Which of the following approaches is the best in terms of runtime for finding the longest path in an unweighted DAG?

**Q8. [4 marks]** Let G be a directed graph composed of only two strongly connected components, A and B.
There is only one edge from a vertex in one SCC to a vertex in the other, which is a->b where a ∈ A and b ∈ B.
Now construct an augmented graph G' by adding three new vertices p, q, and r and the following additional **directed** edges:

- p -> a' for some a' ∈ A
- b' -> q for some b' ∈ B
- p -> r, r -> q and q -> p
- b'' -> p for some b'' ∈ B

How many SCC(s) do/does G' have?

**Q9. [4 marks]**

Suppose G = (V, E) is a connected, undirected, weighted graph. There exists T, a shortest-path spanning tree of G, rooted at source vertex s.
Pick an edge (u, v) in G, and decrease the weight by a constant C > 0. The edge (u, v) can now have a negative weight.
Which of the following **must be true** for the new shortest-path spanning tree T'?

**Q10. [4 marks]** Referring to the implementation of Floyd-Warshall algorithm in the lecture notes, after completing iteration k=3 (i.e. after using the first 3 vertices as intermediates) for a graph with > 4 vertices, the state of the distance matrix is best described by which option?

## Analysis: 21 marks, 3 Questions – 7 marks each

**Q11. [7 marks]** It is impossible to have a binary tree containing unique integers with more than 1 node that is both an AVL tree and a valid binary max heap.

a) True / False?

b) Explain your answer.

**Q12. [7 marks]** A UFDS, as implemented in lectures using both union-by-rank and path compression, has 5,000 elements. Within the UFDS, 2 of the disjoint sets are **A** and **B**:

Set **A** has 1,000 elements with the same representative

Set **B** has only 10 elements, and its representative has a rank of 3

**Claim**: It is possible that unionSet(**A**, **B**) places the representative of set **A** under the representative of set **B**.

a) Is the claim True/False?

b) Give your rationale for your answer to (a).

**Q13. [7 marks]** A weighted connected undirected graph **G** is made up of 2 trees **A** and **B**, as well as 4 additional edges. This means that, if **A** and **B** contain **N** and **M** vertices respectively, then **G** contains **N**+**M** vertices and exactly **N**+**M**+2 edges.

Each of the 4 additional edges (**u**, **v**) is such that **u** is a vertex in **A**, while **v** is a vertex in **B**.

Trees **A** and **B** each contains at least 4 vertices, and do not share any vertices between them.

The weights of ALL the **N**+**M**+2 edges in **G** are distinct.

**Claim**: The Minimum Spanning Tree of **G** will include **exactly one** of the 4 additional edges mentioned above.

a) Is the claim True/False?

b) Give your rationale for your answer to (a).

## Structured: 39 marks, 3 Questions – 13 marks each.

**14) [13 marks]** Range of orange

Ivan has **N** (which can be very large) orange slices, each having a volume (positive floating point number with no fixed precision). Ivan takes out a sharp knife and makes **C** cuts on the oranges.

When Ivan makes a cut, he takes the slice with the largest volume and splits it into two equal volumes (which add up to the original slice's volume). After each cut, you are to output the range of the oranges' volume, i.e. $V_{largest\_slice} – V_{smallest\_slice}$.

You are given the integer **C**, as well as a nonempty array **A** which contains the **N** volumes, one for each orange slice.

Example: **C** = 5, **A** = [8.0, 4.0, 6.0]

the 5 outputs in sequence are: 2.0 1.0 2.0 2.0 1.0

because

| | |
|---|---|
| after 0 cuts: **8.0** 4.0 6.0 | range : 4.0 but not in output |
| after 1 cuts: 4.0 4.0 4.0 **6.0** | range : 2.0 |
| after 2 cuts: **4.0** 4.0 4.0 3.0 3.0 | range : 1.0 |
| after 3 cuts: **4.0** 4.0 3.0 3.0 2.0 2.0 | range : 2.0 |
| after 4 cuts: **4.0** 3.0 3.0 2.0 2.0 2.0 2.0 | range : 2.0 |
| after 5 cuts: 3.0 3.0 2.0 2.0 2.0 2.0 2.0 2.0 | range : 1.0 |

Design a most efficient algorithm to output the **C** ranges, stating clearly any data structures required where they matter.

A correct O(**N** + **C** log (**N** + **C**)) algorithm will score full marks, while a O((**N+C**) log (**N** + **C**)) algorithm will score "high" but not full marks.

**15) [13 marks]** TransferWise, a global money transfer company, needs a system to compute processing fee differences between banks.
Each of the N banks charges a fixed processing fee, but the exact fees are unknown. However, pairwise information about the differences between banks' processing fees are known.
E.g., Bank A charges $5 lower than Bank B.

Processing fees may vary across different pairs of banks, and difference in processing fee may not be constant.
E.g. Bank A can charge $5 lower than Bank B, but Bank C can charge $7 higher than Bank D.

All comparisons between pairs of banks will always lead to a consistent difference in processing fee.
E.g. given two information: Bank A charge $5 lower than Bank B, and Bank B charges $5 higher than Bank C, there will be no conflicting information stating that Bank A and Bank C have different processing fees.

Design a system to answer customer queries about fee differences between any two banks. Assume all queries can be answered based on the given pairwise information.

Sample data, queries and answers are provided below:
Consider 3 banks OSBC, MBS, HCBC along with the following information.

- K1. Transferring through OSBC costs 4 dollars more than transferring through MBS.
- K2. Transferring through OSBC costs 3 dollars less than transferring through HCBC.

**Query 1:** How much more expensive it is to transfer through MBS than OSBC?
**Answer:** -4 dollars. (From K1)

**Query 2:** How much more expensive it is to transfer through MBS than HCBC?
**Answer:** -7 dollars. (-4-3, where -4 is because of K1 and -3 is because of K2)

Ensure you clearly indicate and answer these three parts:

1. Model the given scenario as a graph.
2. Assume the number of pairwise information available is N-1, design an algorithm to answer Q queries efficiently.
3. Assuming the number of pairwise information is available is M, where M is much higher than (N-1), design an algorithm to answer Q queries efficiently.

16) **[13 marks]** Static vs current electricity

Tom has **K** (which can be very large) steel plates numbered 0 to **K**-1 (in that order) in a line from left to right. All plates do not touch each other. Initially, a plate is connected to either one power source or to nothing.

There are only two power sources – a positive source, and a negative source. The positive source always remains positive, and the negative source always remains negative.

Subsequently, wires are secured between 2 plates so that electricity can pass between them (need NOT be adjacent plates):

- **Current electricity flows** along paths from the positive to the negative source
- However, if there is a path to only one source (but not both), then **current does NOT flow**, but the plate becomes positively or negatively charged instead (static electricity)

Hence, each plate takes one of four states:

N Neutral – Plate is not in any way connected to any power source
+ charged – The only source that the plate is connected to is the positive source
- charged – The only source that the plate is connected to is the negative source
C Current – There is current flowing through **some part of the circuit this plate is connected** to (to be precise, current may not be flowing through this plate itself)

Design an algorithm for each of the 2 operations:

`Init(int` **`K`**`, int[]` **`P`**`, int[]` **`N`**`)` – There are **K** plates. **P**, **N** are lists of plate numbers that are connected to the positive, negative source respectively. Perform any initialization on data structures to support the `Wire` operations

`Wire(int` **`L`**`, int` **`R`**`)` – Secure a wire between plate **L** and plate **R** and **return the state of the plates** on both sides of the wire after it is secured (Think about it, the states of both plates will always be the same after the wire is secured…)

For full credit, both operations should run correctly, and the `Wire` operation should run as efficiently as possible ("not far from O(1) time" on average) while the `Init` operation should support the given implementation of `Wire` as efficiently as possible.

Example:

Init(6, {1}, {5, 4}) <-- the metal plates have states ['N', '+', 'N', 'N', '-', '-'] in sequence
Wire(3, 2) returns 'N' <-- states are still ['N', '+', 'N', 'N', '-', '-'] though wire secured
Wire(1, 2) returns '+' <-- states are now ['N', '+', '+', '+', '-', '-'] as plates 2,3 are + charged
Wire(3, 4) returns 'C' <-- states are now ['N', 'C', 'C', 'C', 'C', '-'] as plates 1 to 4 are connected to a circuit with current flowing through
Wire(0, 1) returns 'C' <-- states are now ['C', 'C', 'C', 'C', 'C', '-'] as plate 0 is ALSO connected to a circuit with current flowing through, **even though current doesn't take a path through plate 0**

Tip: It may be useful to implement this helper operation and call it where necessary - FindNewState(A, B) which returns the new state after securing a wire to two plates, where A and B are the old states of the two plates