INSTRUCTIONS

- You must answer ALL the questions.
- Answers to each question should be in the space DIRECTLY BELOW the questions and (if required) on the blank page overleaf of each question.

General exam technique

Do not throw marks away by **NOT** attempting all questions. Suppose you get 7/10 on a question for a 20 minutes effort. Spending another half hour on the same question gets *at most* 3 more marks. On the other hand, were you to spend that time on a new question, you might get another 10 marks.

Do not write un-necessarily long answers. This wastes your valuable exam time. The question will specifically ask for the information required. Do not include the information that is not specifically asked for. If asked to justify your answer, provide a clear, logical and concise reasoning.

You do not have to attempt the questions in order. Some questions require less work but may be worth more marks. Carefully read the paper to decide the order in which you should attempt the questions.

Several questions ask of you to write Pseudocode. Pseudocode is essentially a **high-level description** of a program that should allow a human to understand when it is read. If you feel more comfortable using Python syntax, you are welcome to use it but don't get bogged down with syntax. What is essentially being assessed for such 'pseudocode' questions is your basic understanding and the logic of the algorithm (and not its syntactical correctness).

Best of Luck!

Do not write anything in this table. It is for office use only.

Question	Points	Score
1	10	
2	5	
3	11	
4	4	
5	8	
6	6	
7	10	
8	6	
Total:	60	

This page is intentionally left blank. If needed, you can use this space to write answers.

1. This question is composed of four short questions. Write your answers to each of these questions in no more than a few lines.

   (a) (2 marks) What is the average-case time complexity of Quick Sort algorithm? Briefly justify your answer.

   (b) (3 marks) Assume that we are calling Bellman Ford algorithm **for every vertex** v of the graph to compute all pairs-shortest distances on a weighted graph. What will be the worst-case time complexity of this approach? What will be the worst-case time complexity if the graph is dense? Briefly justify your answer.

This page is intentionally left blank. If needed, you can use this space to write answers.

(c) (3 marks) Consider the flow network shown in Figure 1 where, for each edge label i/j, i indicates the flow value across the edge and j indicates the capacity of the edge. Flow is not shown along the edge if it is 0. What is the min-cut on this flow network and what is its capacity? What is the maximum possible flow in this network?

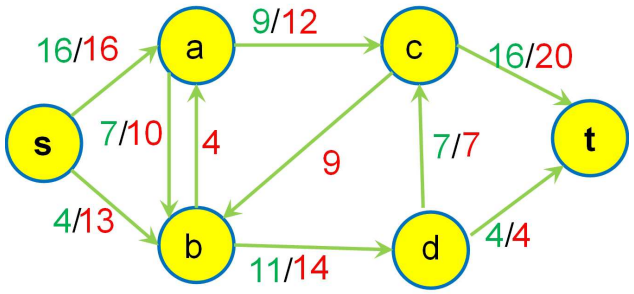

Figure 1: Flow Network

(d) (2 marks) What is the time complexity of Ford Fulkerson algorithm assuming Breadth First Search is used to find augmenting paths? Give a brief reasoning.

This page is intentionally left blank. If needed, you can use this space to write answers.

2. (5 marks) Given a capacity c and a set of items with their weights and values, the 0/1 knapsack problem is to pick items such that their total weight is at most c and their total value is maximized. You **cannot** pick any item more than once. Consider four items A : (9kg; \$550), B : (5kg; \$350), C : (6kg; \$180), and D : (1kg; \$40). If the capacity c is 12kg, the solution to 0/1 knapsack is to pick item A and item D with total weight 10kg and total value \$590.

   Write pseudocode for a dynamic programming algorithm to solve the 0/1 knapsack problem.

5

This page is intentionally left blank. If needed, you can use this space to write answers.

3. (a) (4 marks) Use mathematical induction to prove the following:

$$\sum_{i=0}^{N} ar^i = a + ar + ar^2 + ar^3 + \cdots + ar^N = \frac{a\left(r^{N+1} - 1\right)}{r - 1}$$

This page is intentionally left blank. If needed, you can use this space to write answers.

(b) (5 marks) Write the recurrence relation of time complexity for the `mystery(N)` function shown below and solve it. What is its time complexity in Big-O notation?

```
def mystery(N):
    if N == 1:
        return 1
    else:
        value = 0
        for i in range(2*N):
            value += i
        return value + mystery(N//2)
```

(c) (2 marks) What is the space complexity of `mystery(N)` function given in part(a)? Give a brief justification of your answer.

This page is intentionally left blank. If needed, you can use this space to write answers.

4. (4 marks) Show how the following AVL tree is balanced after **19.5** is added. You need to identify each case (e.g., left-left case) and show how **each** rotation is done. You must also include the balance factors for the nodes in your figures.

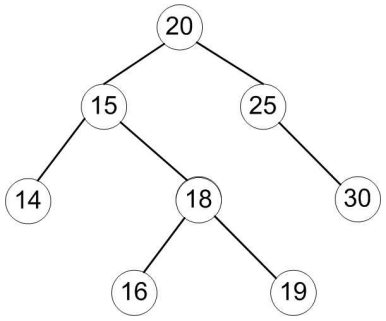

Figure 2: AVL Tree

4

This page is intentionally left blank. If needed, you can use this space to write answers.

5. The Burrows-Wheeler Transform of a string is "ecbba$ga". Answer the following questions.

   (a) (4 marks) Use Last-First columns mapping to reconstruct the original string. The progression of your worked out steps of Last-First columns mapping must be clearly demonstrated in your answer using row numbers and arrows.

This page is intentionally left blank. If needed, you can use this space to write answers.

(b) (4 marks) Using the backwards search strategy, show how you would search the query pattern "ab" and count the number of its occurrences in the original string. Clearly demonstrate the range after processing each character in the query pattern and use this to count the number of its occurrences.

4

This page is intentionally left blank. If needed, you can use this space to write answers.

6. (a) (3 marks) What will be the suffix array of RABBIT? You are not required to show the working – it is sufficient to show ONLY the suffix array.

(b) (3 marks) What are the space and time complexities of constructing a suffix array using the prefix doubling approach when insertion sort is used to sort at each step? Briefly justify your answer.

This page is intentionally left blank. If needed, you can use this space to write answers.

7. (a) (5 marks) Consider the graph G shown in Figure 3. Show how Prim's algorithm computes the minimum spanning tree in graph G by drawing figures after each edge is added to the tree. You will need to draw five figures.

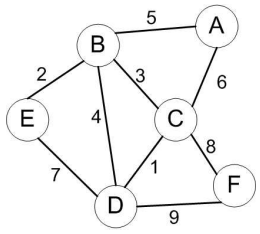

Figure 3: Graph G

5

This page is intentionally left blank. If needed, you can use this space to write answers.

(b) (5 marks) Write the invariant of Prim's algorithm and use it to prove that Prim's algorithm correctly computes a minimum spanning tree.

5

This page is intentionally left blank. If needed, you can use this space to write answers.

8. Let $G = (V, E, W)$ be a **weighted undirected graph**, with the vertex set V, the edge set E, and the corresponding weights set W.

   (a) (4 marks) Write pseudocode for the Bellman-Ford algorithm that finds **shortest distances** between a source vertex to every other vertex in the graph.

   (b) (2 marks) What are the worst-case space and time complexities for Bellman-Ford algorithm.

This is the end of the test.