**COMP2009 2022-23 ADE Coursework THREE (6.25%)**
**Wed. 10-MAY-2023**
**Time: 30 minutes.**
**Do not turn over the page until instructed.**
**Answer ALL FOUR questions for a total of 25 marks.**
**Calculators are not permitted.**
**Write your answers on these sheets within the spaces provided. Please write clearly.**

**Write your name & ID in the box below CLEARLY AND IN UPPER CASE LETTERS**

| | |
|---|---|
| **FAMILY NAME:** | |
| **FIRST NAME(S):** | |
| **Student ID number:** | |
| **Signature:** | |
| **Session: (circle one)** | CS-A32:9am CS-A32:10am Other |

**(Also, write your name on each sheet; in case the sheets become separated.)**

Suggest writing no more than a sentence for each "justification" on the first pass – unless have free time at the end to write more. It is usually a better strategy to attempt every question, than to give a long answer to just some of the questions, and nothing on other questions.

Use the last (blank) page if really needed, but please include a pointer to it – "see last page" – if you want it marked.

(Standard policy) If you think something in a question is incorrect, then please just answer the question as it is – but write a short note about anything you think is wrong. The test is distributed over multiple rooms and times and so it is not possible to do corrections "live". If there an error, then this will be taken account of during the marking.

**For completion by markers:**

Total mark (out of 25):

Q1 Q2 Q3 Q4=

## Question 1. Hashmaps using linear probing [8 marks]

a) Consider a hash table of size N= 7, with (a very simple) hash function

**h(k) = (k+3) mod 7**
and using **linear probing.**

Starting from an empty hash table give the results after each stage of the following sequence of 3 insertions.

You can use that: **h(3)=6**, **h(4)=0**, and **h(10)=6**

Show the result after doing insert(3), after starting from an empty table:

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Entry | | | | | | | |

.. and then, insert(4) gives

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Entry | | | | | | | |

.. and then, insert(10) gives

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Entry | | | | | | | |

Note that you do not need to write justifications, just add entries to the tables.

## Question 1. (cont)

b) Using the same hash table and function as in Q1a, and again using linear probing a sequence of operations resulted in the following state of the table:

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Entry | 2 | | | | | 9 | 3 |

You can use that: **h(9)=5**, **h(3)=6**, and **h(2)=5**

The task is then to do remove(9).

State the problem that will arise if the entry for 9 is simply set to be blank (empty):

Briefly describe ONE scheme for handling removals when using linear probing:

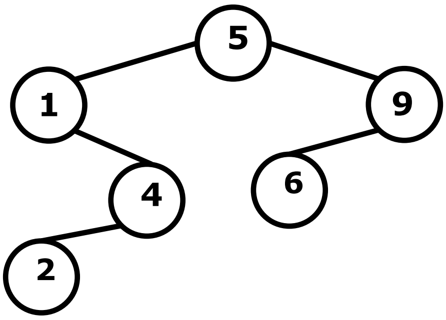
Show the final result of the table after using the scheme you suggested, and then give a **brief** justification:

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| Entry | | | | | | | |

Justification:

## Question 2. Binary Search Trees (BSTs) [6 marks]

Consider the following BST:

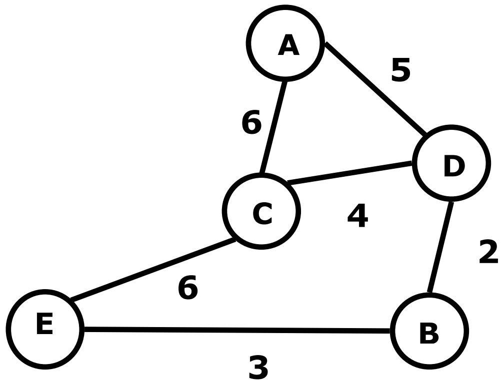



You are to **delete(5)** (i.e. remove 5 from the set of keys) using the method as given in the lectures.

Show and **briefly explain** the process, and give the BST that results after the deletion:

## Question 3. Minimum Spanning Trees (MSTs) [5 marks]

Consider the following graph:




Use Prim's algorithm to find a Minimum Spanning Tree (MST), using **node D as the starting point**. **Show and briefly justify your working**, by simply completing the following table to show the order that edges are added to the tree. (You are given that the first edge to add is D-B with cost 2).

| Iteration | Edge added | Cost | Brief Justification |
|---|---|---|---|
| 1 | D - B | 2 | Shortest edge between D-A, D-C, and D-B |
| 2 | | | |
| 3 | | | |
| 4 | | | |

## Question 4. Change giving **[6 marks]**

Consider a set of coins {1,2,5} and a target of change of 7.
Consider the algorithm of the lectures based on an array A, with meaning:

A[i]=T iff a change of i is possible using the coins considered so far

Complete the following tables to show the state of the array after every iteration. In the arrays, leave the entry as blank to signify "F" for false, and just add the "T" for true – meaning that the sum given by the index (given in the top row) is achievable using the coins so far.

Starting array:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| T | | | | | | | |

After adding coin '1':

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| T | | | | | | | |

After adding coin '2':

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| T | | | | | | | |

After adding coin '5':

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| T | | | | | | | |

From the final array:
Is it possible to give a change of 7? Yes / No (circle one)

Is it possible to give a change of 4? Yes / No (circle one)

**END**