COMP2009 ADE CW3 Answers and Feedback Watch on Moodle in case of updates

QUESTION 1 [8 MARKS]

ON THIS QUESTION THE AVERAGE MARK (of those taking the test) WAS 82%

Q1A. (3 MARKS TOTAL)

1.1.1 Intended Answer

Starting with an empty table of length 7, and a hash function that maps as $\{3 \to 6, 4 \to 0, 10 \to 6\}$, insert the values 3, then 4, then 10. After each insertion, the new table should be given as:

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Entry							**3**

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Entry	**4**						**3**

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Entry	**4**	**10**					**3**

ERRORS:
Some seemed to remove the previous entries after each new insertion, but the point was they are done sequentially – otherwise it will not show any effects of the linear probing.

Q1b. (5 marks total)

1.1.2 Answer

<u>Statement of problem of removal</u>
Identify and explain that just removing an element when using linear probing can cause entries to be undiscoverable, "lost", as linear probing stops searching for elements once a blank is found.
In particular the entry '2' would be lost – that is, doing get(2) would fail even though it is still supposed to be in the table.

ERRORS:
Being too vague – an not mentioning that specifically get(2) will fail
Saying that get(3) will fail – which is incorrect as '3' is at its 'home' in cell index 6 – as h(3) = 6 – and so would be found.

<u>Removal scheme</u>

Two options

1. Mark as removed/lazy deletion/tombstone e.g. with "D", but any symbol is okay as long as it is different from the "blank" in the other cells.
2. Remove and reinsert all entries to the right of what was just removed.

Result and justification

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Entry	2					D	3

The 9 is removed from [5] and [5] **marked** as removed/deleted

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**
Entry						2	3

- The 9 is removed from [5].
- All elements to the right are removed ([6] and [0]) until a blank is found ([1]).*
- The removed elements 3 and 2 are then reinserted into indices [6] and [5] respectively.

*can be done one-by-one or all at once.
Other options were also allowed.

ERROR:
A few put the same symbol for deleted as for blank and this would not work

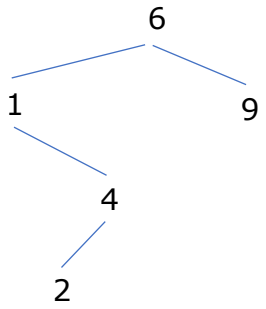
QUESTION 2 [6 MARKS]

ON THIS QUESTION THE AVERAGE MARK (of those taking the test) WAS 59%

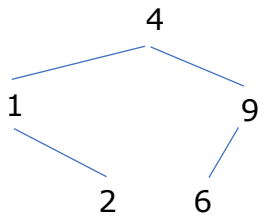
ANSWER

Same as given in the lectures.
1. Find the next or previous entry to '5' that would occur in an in-order traversal – which is 4 or 6
2. COPY the entry of either 4 or 6 and OVERWITE '5'
3 Delete which ever node of 4 or 6 was selected

Results in correct BST: From using "next"=6



Or from using "prev"=4



ERRORS
Just moving 6 into the place of 5. This happens to be okay in this case, but was not the standard method.

Not mentioning that have to remove the prev/next that was used.

Just doing something that moves around the nodes in a way that is not systematic, and just happens to work. It was not enough to just end up with one of the trees above, but had to be done in a way that would be programmable, in same way as lectures, and not just ad hoc.

Giving a final answer that was not a BST.

QUESTION 3

ON THIS QUESTION THE AVERAGE MARK WAS 53%

ANSWER

Iteration	Edge Added	Cost -OR- cost so far	Justification
1	D-B	2	Shortest edge of D-A, D-C, and D-B.
2	B-E	3 (or 5)	Shortest edge of D-A, D-C, and B-E.
3	D-C	4 (or 9)	Shortest edge of D-A, D-C, and E-C.
4	D-A	5 (or 14)	Shortest edge of D-A, E-C, and C-A.

No other alternatives are possible when following Prim's algorithm.

COMMON ERROR – follow a path, by adding only from last added node, this gives
D-B
B-E ok
E-C wrong
C-A wrong

(For those doing AIM, this might be recognised as "nearest-neighbour" heuristic for TSP)

QUESTION 4 [6 MARKS]

ON THIS QUESTION THE AVERAGE MARK WAS 69%

Answer

After considering coin of value 1:

0	1	2	3	4	5	6	7
T	T						

(1 mark)

After considering coin of value 2:

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

(1 mark)

After considering coin of value 5:

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

Could also result in the following if they did not "return" after finding the target value of 7.

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

Is it possible to give a change of 7?

Yes

Is it possible to give a change of 4?

No (or I suppose "maybe" should be the right answer if we return immediately after finding 7)?

NOTE:
The algorithms given in lectures would stop on success, and so the entries for indices 5,6,7 might be
… F F F
or
… F F T
and these were also acceptable.

But no algorithm would give … T F T or … F T F so this was considered an ERROR.