Grade 11 Computing

Unit 3 · Data Structures

ANSWER KEY · FOR TEACHERS

Full marks 33

Give the marks for the points in the marking scheme; an answer in other words that says the same is right.

1 Key words

1. stack
2. queue
3. linked list
4. node
5. hash table
6. collision
7. binary search tree
8. heap

Each carries 0.5 mark. **K · 4**

2 Label the diagram

1. Dawa
2. Hari
3. Pemba
4. Sita

Each carries 0.5 mark. **A · 2**

3 Multiple choice

i. **a.** 3 **A · 1**

After push 3, push 8: [3, 8]. Pop takes 8: [3]. Push 5, push 1: [3, 5, 1]. Two pops take 1, then 5: [3]. So 3 is on top (and alone).

ii. **c.** nothing: the forward stack was emptied when E was opened **A · 1**

Back twice moved D and then C onto the forward stack, leaving you on B. Opening a new page, E, pushes B onto the back stack and empties the forward stack, so C and D can no longer be reached by Forward.

iii. **c.** 7 **U · 1**

All seven keys are in one bucket's list, so a look-up may have to check every one of them: the hash table has become a linear search. A good hash function spreads the keys so each bucket holds only one or two.

iv. **c.** 5 **A · 1**

Each number is bigger than all before it, so each goes to the right of the last: the tree is a single line, 10 → 20 → 30 → 40 → 50. Finding 50 compares with all five. Added in the order 30, 10, 40, 20, 50, the same numbers make a tree three levels high.

v. **b.** about 10 **U · 1**

The new patient sifts up at most from the bottom level to the root. A complete tree of 1,000 nodes has 10 levels ($2^{10} = 1{,}024$), so at most about 9 or 10 swaps: O(log n).

4 What does it print?

MARKING SCHEME **A · 3**

- ✓ `Hetauda Pokhara`: the stack gives back the city added last, the queue the city added first.
- ✓ `Janakpur Dharan Butwal`: Janakpur, just pushed, comes off the stack first, then Dharan; the queue's front is now Butwal.
- ✓ `2 ['Dharan', 'Hetauda', 'Janakpur']`: Pokhara and Butwal are left on the stack; the queue has lost its first two and gained Janakpur at the back.

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

5 Trace the program

ANSWER **A · 4**

ch	stack after it	Output
{	['{']	
[	['{', '[']	
(	['{', '[', '(']	
)	['{', '[']	
]	['{']	
}	[]	
(	['(']	
]	[]	
		False

The last] pops (, the wrong kind, so the function gives back False straight away.

MARKING SCHEME

- ✓ The three openers pushed in order, and),], } each popping its own kind, leaving the stack empty.
- ✓ (pushed again after the }.
- ✓ The last] pops (, which doesn't match.
- ✓ Output False, given back at once (the loop doesn't carry on).

6 Grow a search tree

ANSWER **A · 4**

i. 40 is the root. Its left child is 25, with children 10 and 30, and 27 is the left child of 30. Its right child is 60, with children 50 and 70, and 55 is the right child of 50.

```
OUTPUT
            40
          /    \
        25      60
       /  \    /  \
     10    30  50   70
          /      \
        27        55
```

ii. 10, 25, 27, 30, 40, 50, 55, 60, 70: sorted order.

iii. 27 takes **4** looks: 40, 25, 30, 27. 35 takes **3**: 40, 25, 30, and 30 has no right child, so 35 isn't there.

MARKING SCHEME

- ✓ 25 and 60 under 40, and 10, 30, 50, 70 under them in the right places.
- ✓ 27 as the left child of 30, and 55 as the right child of 50.
- ✓ In-order: 10, 25, 27, 30, 40, 50, 55, 60, 70.
- ✓ 4 looks for 27, 3 for 35.

7 Find and fix the bug

MARKING SCHEME **A · 3**

- ✓ An AttributeError on line 8.
- ✓ The loop never checks for the end of the list: after the last node, node is None.
- ✓ `while node is not None and node.value != target:` (the check for None first).

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking

8 Short answer

MARKING SCHEME U · 2

- ✓ i. A heap (a priority queue): the most urgent is always at the front, and adding or taking a patient takes only about $\log_2 n$ swaps.
- ✓ ii. A binary search tree: a name is found in a few looks, and an in-order traversal lists the names in alphabetical order (a dictionary finds a name but can't list them in order).

9 Write a program

ANSWER A · 4

```
def height(node):
    if node is None:
        return 0
    return 1 + max(height(node.left), height(node.right))

def leaves(node):
    if node is None:
        return 0
    if node.left is None and node.right is None:
        return 1
    return leaves(node.left) + leaves(node.right)
```

For the tree of 40, 25, 60, 10, 30, 50, 70, 27 and 55, `height(root)` gives 4 and `leaves(root)` gives 4 (10, 27, 55 and 70).

MARKING SCHEME

- ✓ The base case of each: 0 for None.
- ✓ `height`: 1 plus the bigger of the two subtrees' heights.
- ✓ `leaves`: 1 for a node with no children.
- ✓ `leaves`: otherwise the leaves of the left subtree plus those of the right.

10 Think

ANSWER HA · 2

Both color every square joined to the start, because each takes squares off its list until the list is empty, and every square of the area is put on it from a neighbor; only the order differs. The stack always takes the newest square, so the paint runs on along one path, and the squares beside the path wait on the list (some put on twice) until the snake comes back for them. The queue takes the oldest, so the paint spreads in rings, and only the squares on the edge of the paint wait on the list.

MARKING SCHEME

- ✓ Same squares: each fills everything reachable from the start, in a different order, until its list is empty.
- ✓ The stack follows the newest square, leaving squares waiting all along its path (some twice); the queue spreads in rings, with only the edge of the paint waiting.

11 Activity: Grow a search tree

(not marked)

WHAT STUDENTS SHOULD FIND

In the mixed order, the 15 numbers fit in 4 levels, so any search takes at most 4 looks. In sorted order each number goes to the right of the last: the tree is a single line, 15 levels deep, and finding the biggest number takes 15 looks, as many as a linear search.

Marks by level	Knowledge	Understanding	Application	Higher ability	Total
Marks	4	4	23	2	**33**

Science Playground, Katunje, Bhaktapur
Book a class visit: https://science-playground.com/booking