



QUESTION STYLES

Faculty of Applied Science
Department of: Cybersecurity
Course: Data Structures & Algorithms

Theoretical

Q. Choose the correct answer.

1. What is the output of this code?

```
mylist = [8, 5, 2, 9]
mylist.insert(1, 7)
mylist.pop()
print("The list after modification:", mylist)
```

A. [7, 5, 2, 9]

B. [8, 7, 5, 2]

C. [8, 5, 2, 9]

D. [5, 2, 9, 7]

2. The module in Python contains additional data structures, which are more beneficial than the built-in data structures in certain situations.

A. math

B. random

C. collection

D. time

3. Which one of the following statements is correct?

A. Stack is a non-linear data structure.

B. `append()` is a method used to add a single element at the beginning of a list.

C. Elements in Queue are inserted at the rear and removed from the front of queue.

D. Bubble sort is an unstable sorting algorithm.

Q. Fill in the blank with a related Python statement.

```
class Queue:
```

```
    def __init__(self):
```

```
        self.items = [ ]
```

```
    def isEmpty(self):
```

```
        return .....
```



QUESTION STYLES

Faculty of Applied Science
Department of: Cybersecurity
Course: Data Structures & Algorithms

Q. Fill in the blank with related Python statement.

```
class Stack:  
  
    def __init__(self):  
        self.items = [ ]  
  
    def isEmpty(self):  
        return self.items == [ ]  
  
    def push(self, newItem):  
        .....  
  
    def peek(self):  
        return .....
```

Q. Write down the output of the block of code below.

```
S = Stack()  
S.push('a')  
S.push('b')  
S.push('c')  
S.pop()  
print(S.size())
```

Q. Write down the output of the block of code below.

```
List1 = [ ]  
for i in range(7):  
    List1.append(i)  
List1.remove(5)  
print(List1)
```



QUESTION STYLES

Faculty of Applied Science
Department of: Cybersecurity
Course: Data Structures & Algorithms

Q. Find the total cost ($T(n)$) and big-O notation of the algorithm below.

	Cost	Time Complexity (Big-O Notation)
sum = 0	$c1 * 1$	$O(1)$
for i in range(N):		
for j in range(N):		
sum += i*j	$c2 * (N^2)$	$O(N^2)$

$$\text{Total Cost: } T(n) = c1 + c2 * (N^2)$$

$$\text{Big-O Notation} = O(1) + O(N^2) = O(N^2)$$

Q. Find the total cost ($T(n)$) and big-o notation of the algorithm below.

	Cost	Time Complexity (Big-O Notation)
List1 = []	$c1 * 1$	$O(1)$
for i in range(7):		
List1.append(i)	$c2 * 7$	$O(1)$
List1.remove(5)	$c3 * 1$	$O(1)$
print(List1)	$c4 * 1$	$O(1)$

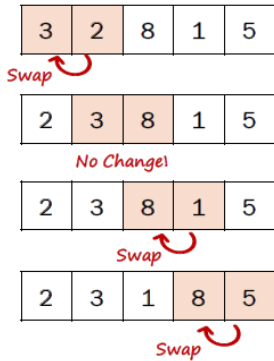
$$\text{Total Cost: } T(n) = c1 + c2 * (7) + c3 + c4$$

$$\text{Big-O Notation} = O(1) + O(1) + O(1) + O(1) = O(1)$$

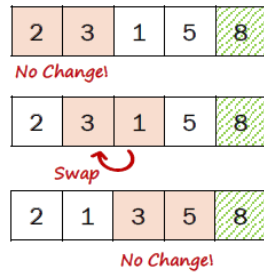
Q. Illustrate how the bubble sorting algorithm sorts the following list of numbers. (write all steps)

3	2	8	1	5
---	---	---	---	---

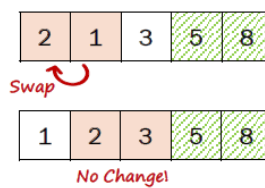
Round 1



Round 2



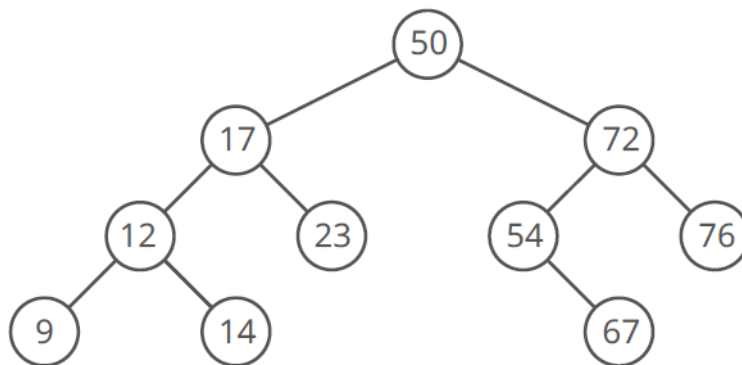
Round 3



Round 4

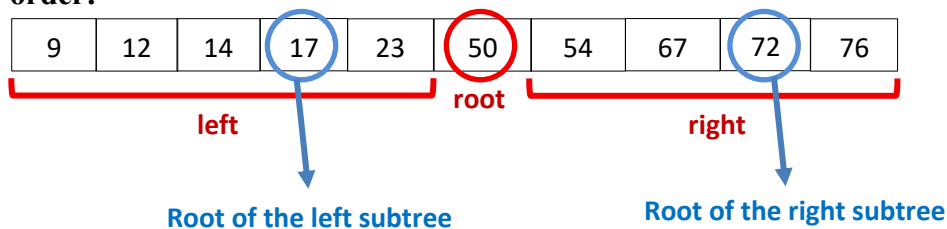


Q. Considering the following BST, write all DFS traversals (in-order, pre-order and post-order) of the tree.



DFS Traversals:

1. In-order:

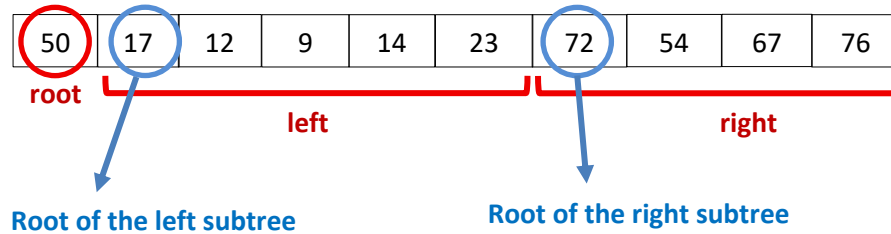




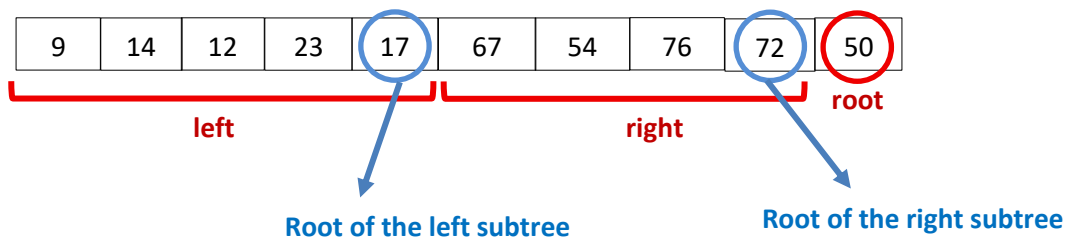
QUESTION STYLES

Faculty of Applied Science
Department of: Cybersecurity
Course: Data Structures & Algorithms

2. Pre-order:



3. Post-order:





QUESTION STYLES

Faculty of Applied Science
Department of: Cybersecurity
Course: Data Structures & Algorithms

Practical

Q. Write a Python function using stack to check for balanced parentheses.

- The parentheses are balanced if:
 - Every opening parenthesis has a corresponding closing parenthesis of the same type.
 - The pairs of parentheses are properly nested.

```
def check_balanced_parantheses(myStr):  
    stack = []  
    for symbol in myStr:  
        if symbol in '({[':  
            stack.append(symbol)  
        elif symbol in ')}]':  
            if (not stack or  
                (symbol == ')' and stack[-1] != '(') or  
                (symbol == '}' and stack[-1] != '{') or  
                (symbol == ']' and stack[-1] != '[')):  
                print(False)  
                break  
            stack.pop()  
        else:  
            print(True if not stack else False)  
  
check_balanced_parantheses('{([[]])}')  
check_balanced_parantheses('{([)]}')
```



QUESTION STYLES

Faculty of Applied Science
Department of: Cybersecurity
Course: Data Structures & Algorithms

Q. Implement the Queue data structure by defining a class for it. Consider only the following queue methods.

- isEmpty() →
- enqueue() →
- dequeue() →

Then create an object of the queue and do the following operations:

- A.** Enqueue these elements to the queue: $10 \rightarrow 50 \rightarrow 30 \rightarrow 20$
- B.** Dequeue an element from the queue.
- C.** Dequeue an element from the queue.

```
class Queue:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []

    def enqueue(self, newItem):
        self.items.append(newItem)

    def dequeue(self):
        return self.items.pop(0)
```

```
# Creating a Queue object
queue = Queue()
```

```
# Calling object methods
queue.enqueue(10)
queue.enqueue(50)
queue.enqueue(30)
queue.enqueue(20)
```

```
# Removing two elements from the queue
queue.dequeue()
queue.dequeue()
```