Text Areas : PlainText

Possible Answers :

4

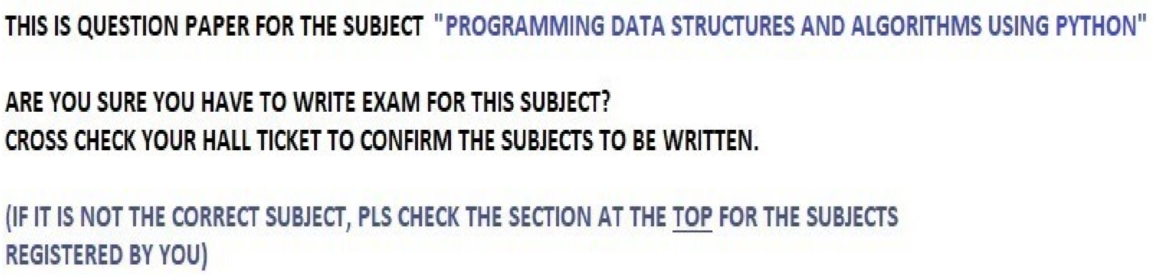
PDSA

Section Id :	64065321949
Section Number :	6
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065349203
Question Shuffling Allowed :	No

Question Number : 88 Question Id : 640653346885 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question



THIS IS QUESTION PAPER FOR THE SUBJECT "PROGRAMMING DATA STRUCTURES AND ALGORITHMS USING PYTHON"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?
CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531153121. ✓ YES

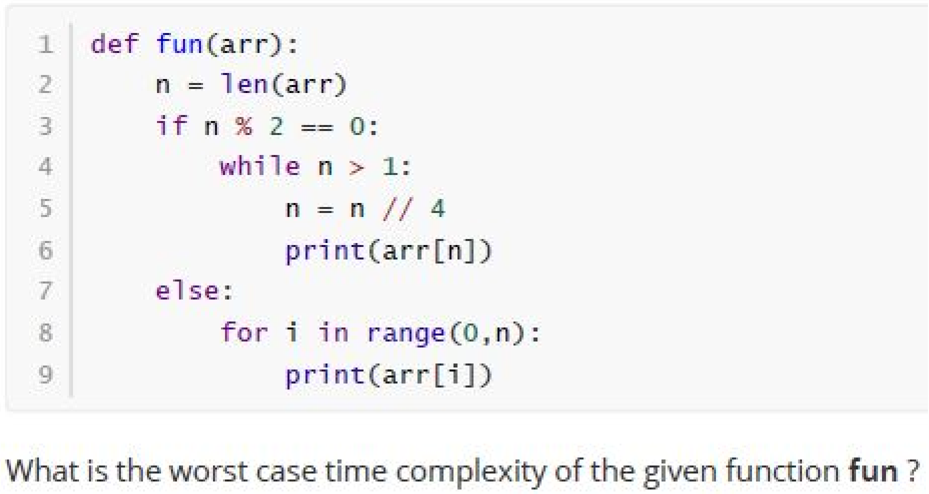
6406531153122. ✗ NO

Sub-Section Number :	2
Sub-Section Id :	64065349204
Question Shuffling Allowed :	Yes

Question Number : 89 Question Id : 640653346886 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question



```
def fun(arr):
    n = len(arr)
    if n % 2 == 0:
        while n > 1:
            n = n // 4
            print(arr[n])
    else:
        for i in range(0,n):
            print(arr[i])
```

What is the worst case time complexity of the given function **fun** ?

Options :

6406531153123. ✓ $O(n)$

6406531153124. ✗ $O(n^2)$

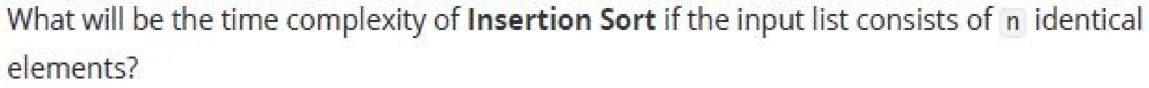
6406531153125. ✗ $O(logn)$

6406531153126. ✗ $O(n^4)$

Question Number : 90 Question Id : 640653346887 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 2

Question Label : Multiple Choice Question



What will be the time complexity of **Insertion Sort** if the input list consists of `n` identical elements?

Options :

6406531153127. ✗ $O(\log n)$

6406531153128. ✓ $O(n)$

6406531153129. ✗ $O(n \log n)$

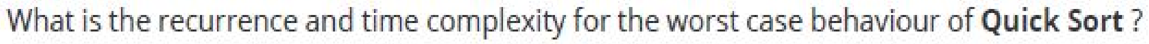
6406531153130. ✗ $O(n^2)$

Sub-Section Number :	3
Sub-Section Id :	64065349205
Question Shuffling Allowed :	Yes

Question Number : 91 Question Id : 640653346888 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question



What is the recurrence and time complexity for the worst case behaviour of **Quick Sort** ?

Options :

6406531153131. ✖ Recurrence is $T(n) = 2T(n-1) + O(n)$ and time complexity is $O(n^2)$

6406531153132. ✔ Recurrence is $T(n) = T(n-1) + O(n)$ and time complexity is $O(n^2)$

6406531153133. ✖ Recurrence is $T(n) = T(n-1) + O(1)$ and time complexity is $O(n)$

6406531153134. ✖ Recurrence is $T(n) = 2T(n/2) + O(n)$ and time complexity is $O(n \, log \, n)$

Question Number : 92 Question Id : 640653346890 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following functions:

$$f1 = n\sqrt{n}$$
$$f2 = \log^2 n$$
$$f3 = n \log n$$
$$f4 = \log \log n$$

Which of the following is correct order of increasing growth rate?

Options :

6406531153139. ✖ $f2 < f4 < f1 < f3$

6406531153140. ✖ $f2 < f4 < f3 < f1$

6406531153141. ✔ $f4 < f2 < f3 < f1$

6406531153142. ✖ $f1 < f3 < f2 < f4$

Question Number : 93 Question Id : 640653346894 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following statements:

1. A single stack can be used to check whether a word is palindrome or not
2. A stack can be used to evaluate expressions.
3. The last element to be inserted into a stack will always be the last element to be taken out of the stack.

Choose the correct option regarding the given statements.

Note: *A palindrome is a word that spells the same from both sides. Eg: radar*

Options :

6406531153155. ✗ Statement 1 and Statement 3 are false

6406531153156. ✗ Statement 2 and Statement 3 are false

6406531153157. ✗ Only statement 1 is false

6406531153158. ✓ Only statement 3 is false

Question Number : 94 Question Id : 640653346895 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

According to the conventional definition, an array is a fixed size data structure whose values are contiguously located in memory, whereas a linked list is a dynamic collection (size can change) of values that are not contiguously located in memory.

Considering the above definitions, if we use a Binary search algorithm to find a value from a linked list, then what would be the worst-case time complexity of Binary search?

Options :

6406531153159. ✖ $O(n \log n)$

6406531153160. ✖ $O(n^2)$

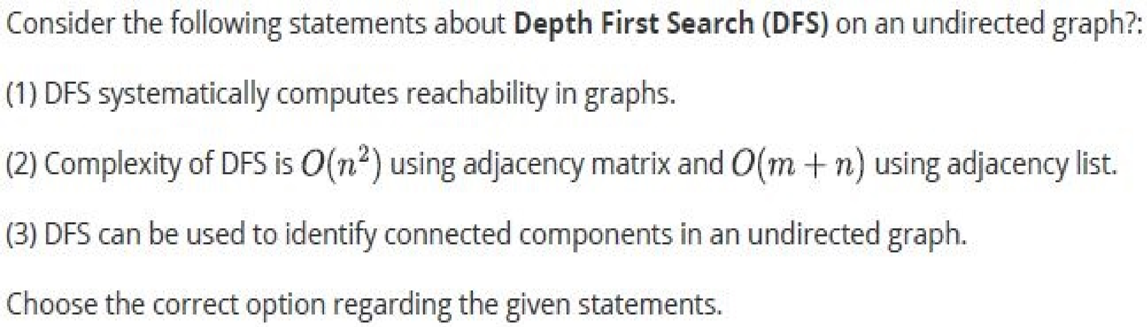
6406531153161. ✔ $O(n)$

6406531153162. ✖ $O(\log n)$

Question Number : 95 Question Id : 640653346896 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question



Consider the following statements about **Depth First Search (DFS)** on an undirected graph?:

(1) DFS systematically computes reachability in graphs.

(2) Complexity of DFS is $O(n^2)$ using adjacency matrix and $O(m+n)$ using adjacency list.

(3) DFS can be used to identify connected components in an undirected graph.

Choose the correct option regarding the given statements.

Options :

6406531153163. ✖ Only Statement 1 and Statement 3 are true

6406531153164. ✖ Only Statement 2 and Statement 3 are true

6406531153165. ✖ Only Statement 1 and Statement 2 are true

6406531153166. ✔ All statements are true

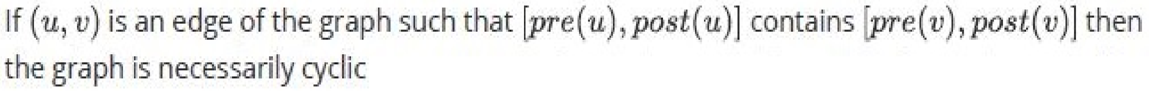
Question Number : 96 Question Id : 640653346900 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

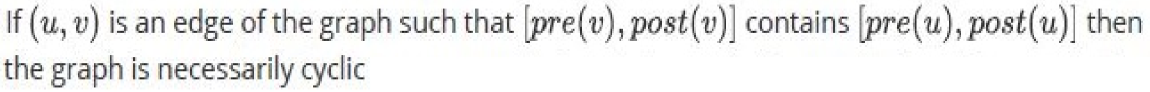
Correct Marks : 3

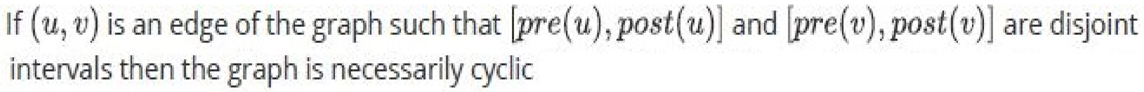
Question Label : Multiple Choice Question

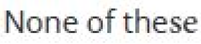
Consider a connected, directed graph on which DFS is executed. Which of the following options are true regarding pre and post numbering used in the DFS algorithm on the graph?

Options :

6406531153178. ✖ 
If (u, v) is an edge of the graph such that $[pre(u), post(u)]$ contains $[pre(v), post(v)]$ then the graph is necessarily cyclic

6406531153179. ✓ 
If (u, v) is an edge of the graph such that $[pre(v), post(v)]$ contains $[pre(u), post(u)]$ then the graph is necessarily cyclic

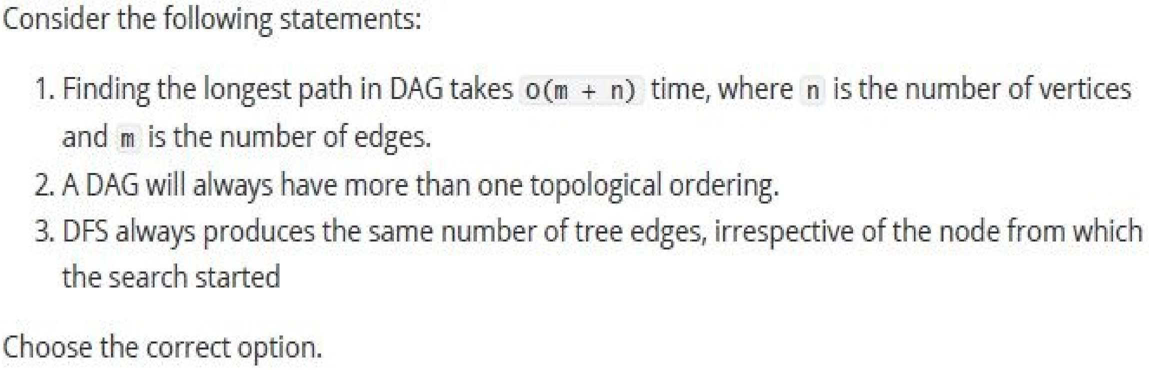
6406531153180. ✖ 
If (u, v) is an edge of the graph such that $[pre(u), post(u)]$ and $[pre(v), post(v)]$ are disjoint intervals then the graph is necessarily cyclic

6406531153181. ✖ 
None of these

Question Number : 97 Question Id : 640653346901 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question


Consider the following statements:

1. Finding the longest path in DAG takes `O(m + n)` time, where `n` is the number of vertices and `m` is the number of edges.
2. A DAG will always have more than one topological ordering.
3. DFS always produces the same number of tree edges, irrespective of the node from which the search started

Choose the correct option.

Options :

6406531153182. ✖ Only statement 1 is true

6406531153183. ✖ Statement 1 and Statement 2 are true

6406531153184. ✓ Statement 1 and Statement 3 are true

6406531153185. ✖ Only statement 3 is true

Sub-Section Number : 4

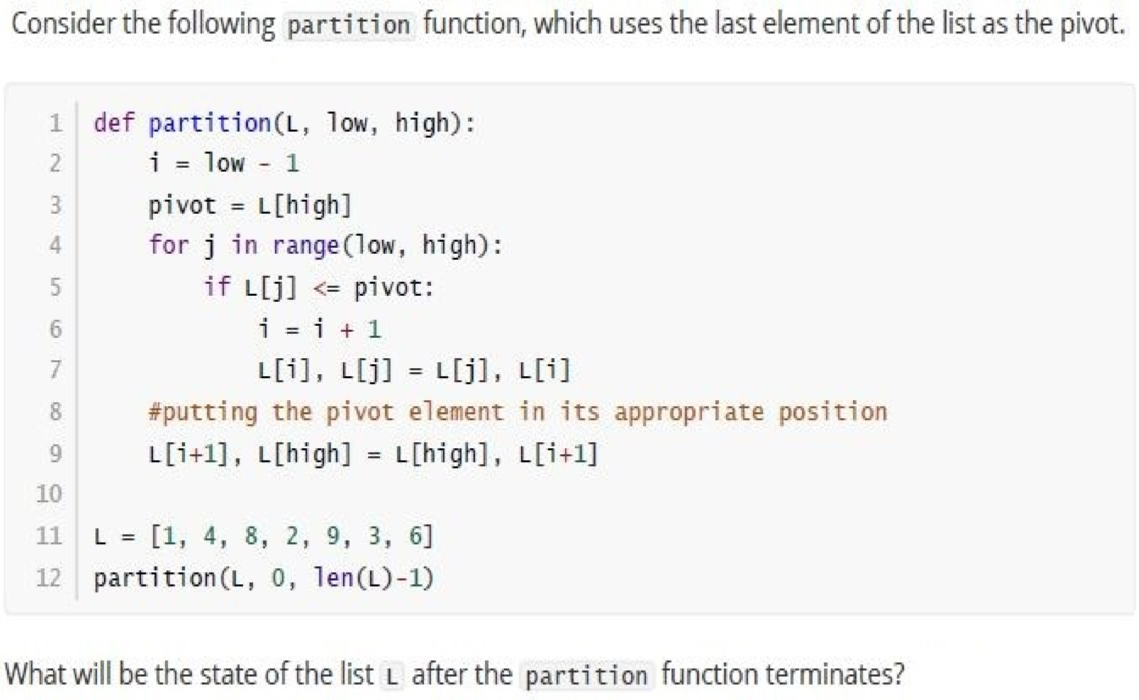
Sub-Section Id : 64065349206

Question Shuffling Allowed : Yes

Question Number : 98 Question Id : 640653346889 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

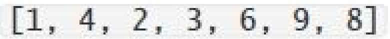


Consider the following `partition` function, which uses the last element of the list as the pivot.

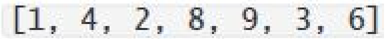
```
def partition(L, low, high):
    i = low - 1
    pivot = L[high]
    for j in range(low, high):
        if L[j] <= pivot:
            i = i + 1
            L[i], L[j] = L[j], L[i]
    #putting the pivot element in its appropriate position
    L[i+1], L[high] = L[high], L[i+1]

L = [1, 4, 8, 2, 9, 3, 6]
partition(L, 0, len(L)-1)
```

What will be the state of the list `L` after the `partition` function terminates?

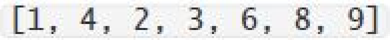
Options :

6406531153135. ✖ 

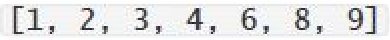
`[1, 4, 2, 3, 6, 9, 8]`

6406531153136. ✖ 

`[1, 4, 2, 8, 9, 3, 6]`

6406531153137. ✓ 

`[1, 4, 2, 3, 6, 8, 9]`

6406531153138. ✖ 

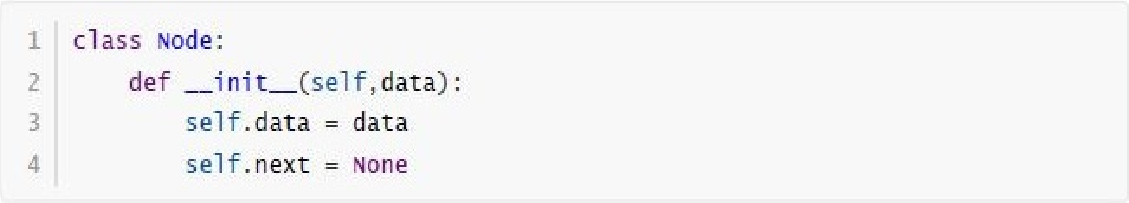
`[1, 2, 3, 4, 6, 8, 9]`

Question Number : 99 Question Id : 640653346891 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

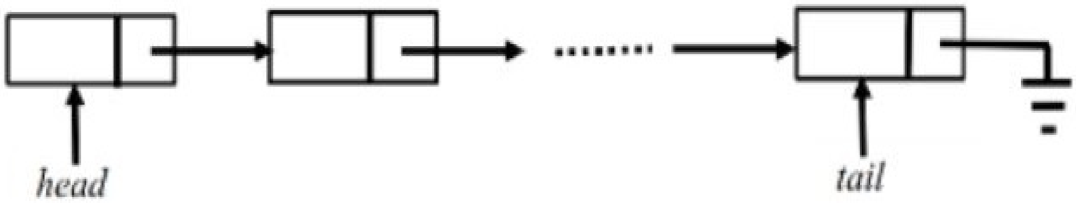
Correct Marks : 4

Question Label : Multiple Choice Question



```
class Node:
    def __init__(self,data):
        self.data = data
        self.next = None
```

Consider an implementation of a singly linked list, where each node is created using the given class `Node`. Suppose it has a `head` pointer that points to the first node of the linked list and a `tail` pointer that points to the last element of the linked list.



Suppose we want to perform the following operations on the given linked list:-

1. Insertion of the new node at the front of the linked list.
2. Insertion of the new node at the end of the linked list.
3. Deletion of the first node of the linked list.
4. Deletion of the last node of the linked list.

Which of the following option represents the correct complexity for each operation?

Options :

6406531153143. ✖ $1 - O(1), 2 - O(n), 3 - O(1), 4 - O(1)$

6406531153144. ✖ $1 - O(1), 2 - O(1), 3 - O(1), 4 - O(1)$

6406531153145. ✖ $1 - O(1), 2 - O(n), 3 - O(1), 4 - O(n)$

6406531153146. ✔ $1 - O(1), 2 - O(1), 3 - O(1), 4 - O(n)$

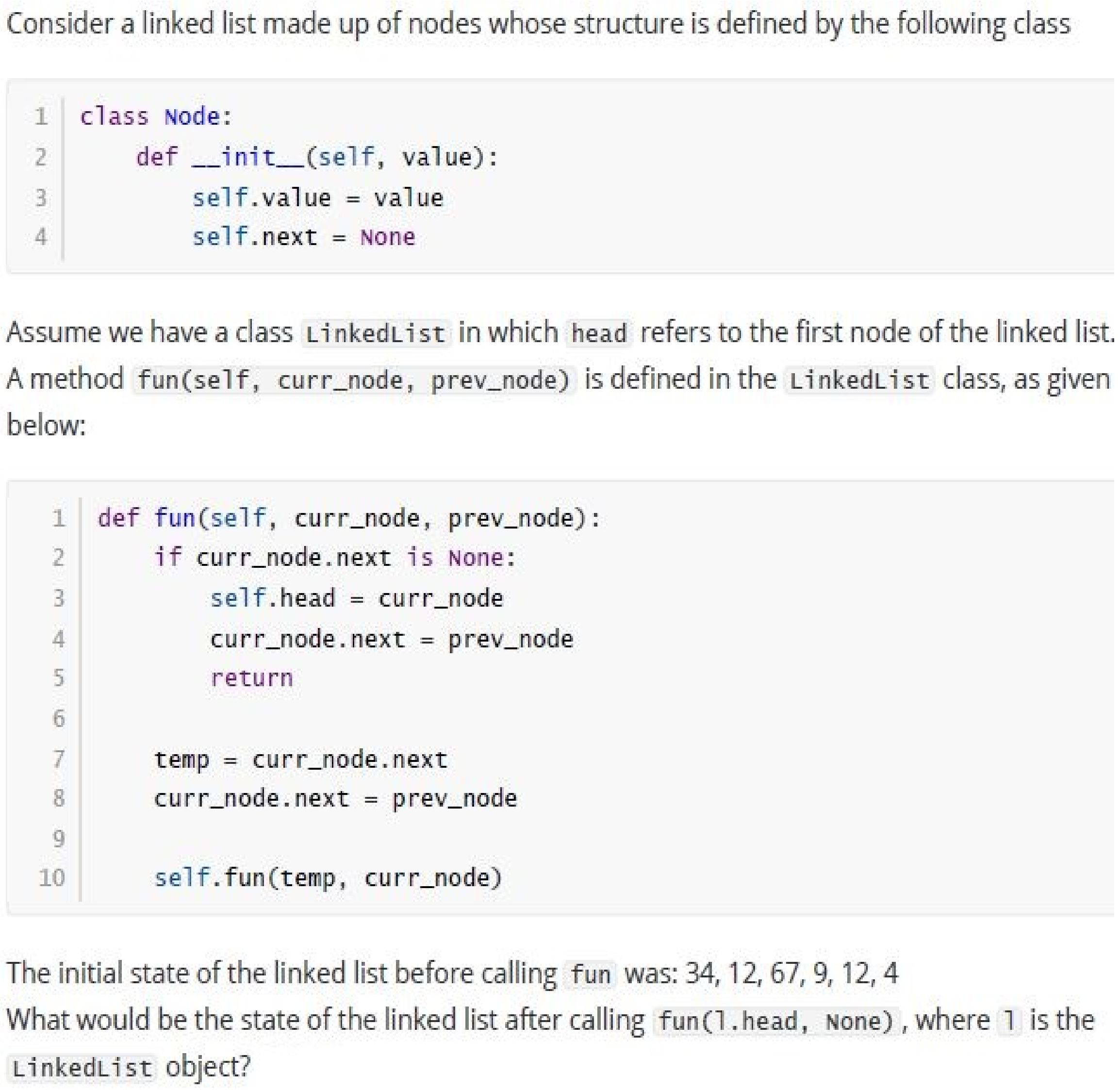
Question Number : 100 Question Id : 640653346892 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question



Consider a linked list made up of nodes whose structure is defined by the following class

```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
```

Assume we have a class `LinkedList` in which `head` refers to the first node of the linked list. A method `fun(self, curr_node, prev_node)` is defined in the `LinkedList` class, as given below:

```
def fun(self, curr_node, prev_node):
    if curr_node.next is None:
        self.head = curr_node
        curr_node.next = prev_node
        return

    temp = curr_node.next
    curr_node.next = prev_node

    self.fun(temp, curr_node)
```

The initial state of the linked list before calling `fun` was: 34, 12, 67, 9, 12, 4

What would be the state of the linked list after calling `fun(l.head, None)`, where `l` is the `LinkedList` object?

Options :

6406531153147. ✗ 12, 34, 9, 67, 4, 12

6406531153148. ✓ 4, 12, 9, 67, 12, 34

6406531153149. ✗ 4, 12, 67, 9, 12, 34

6406531153150. ✗ None of these

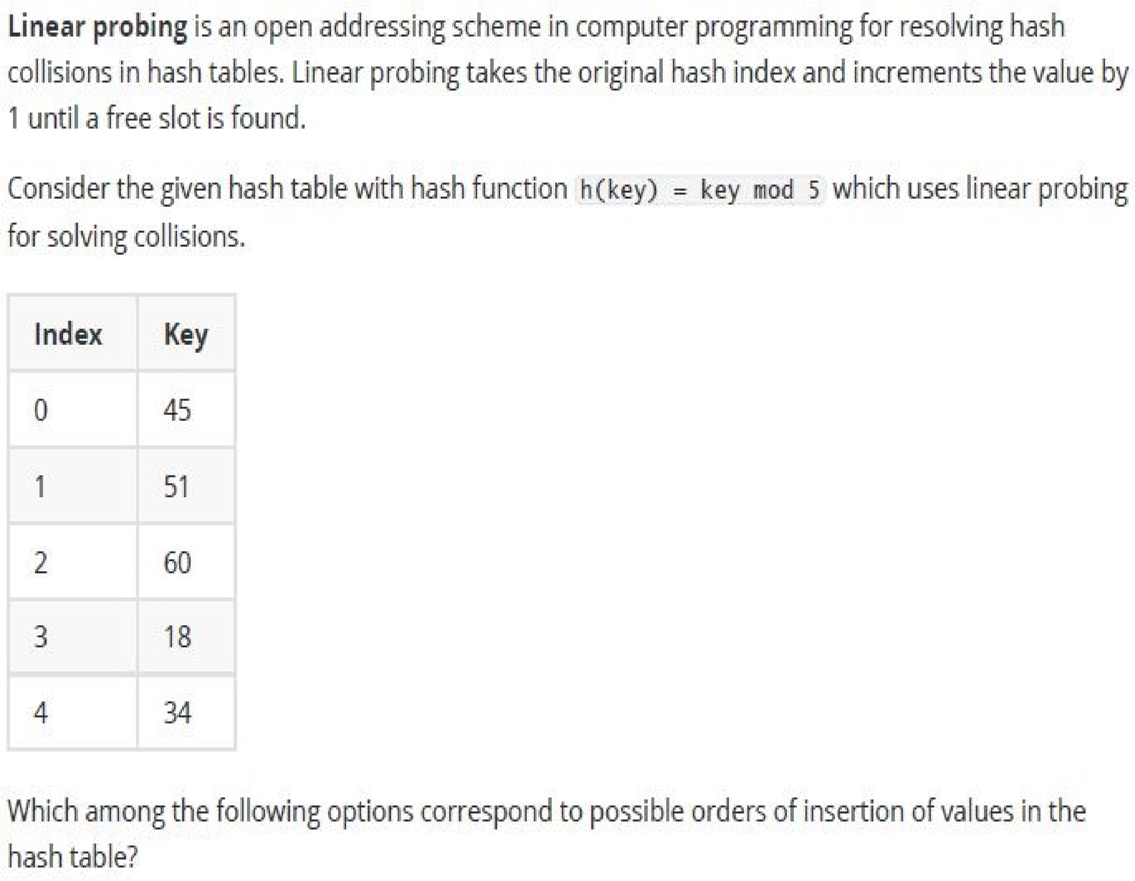
Sub-Section Number :	5
Sub-Section Id :	64065349207
Question Shuffling Allowed :	Yes

Question Number : 101 Question Id : 640653346893 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 3

Question Label : Multiple Select Question



Linear probing is an open addressing scheme in computer programming for resolving hash collisions in hash tables. Linear probing takes the original hash index and increments the value by 1 until a free slot is found.

Consider the given hash table with hash function `h(key) = key mod 5` which uses linear probing for solving collisions.

Index	Key
0	45
1	51
2	60
3	18
4	34

Which among the following options correspond to possible orders of insertion of values in the hash table?

Options :

6406531153151. ✓ 51, 18, 45, 60, 34

6406531153152. ✗ 18, 60, 45, 51, 34

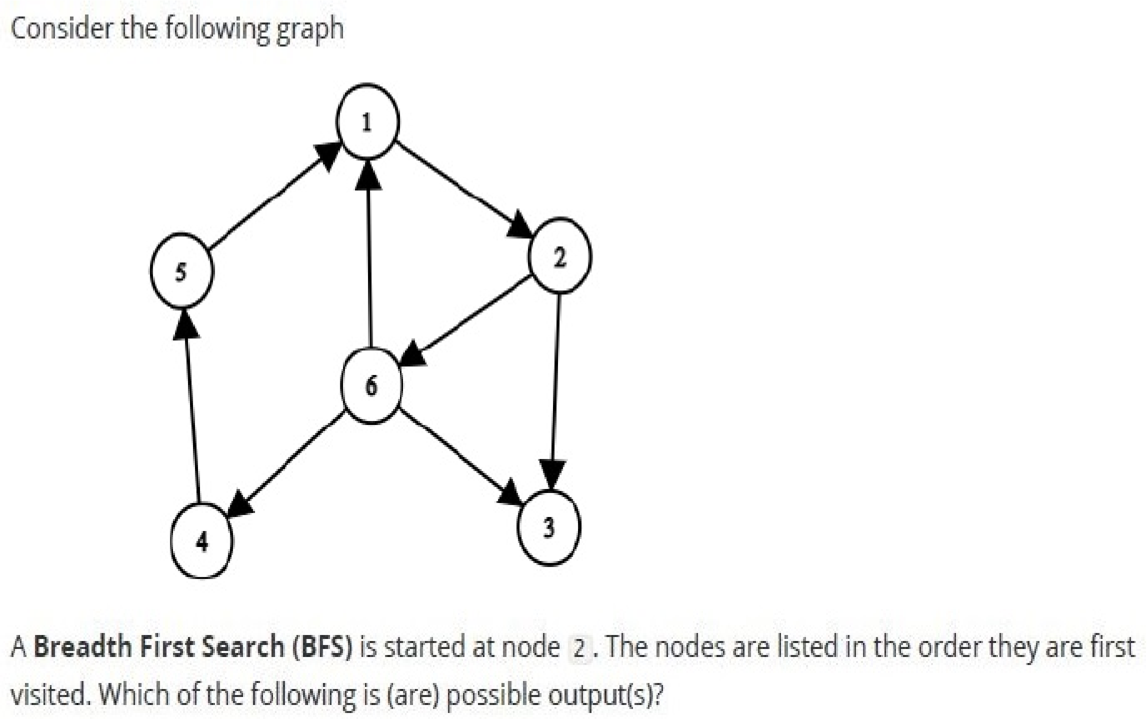
6406531153153. ✗ 18, 45, 34, 60, 51

6406531153154. ✓ 34, 45, 18, 51, 60

Question Number : 102 Question Id : 640653346897 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Select Question



Consider the following graph

A **Breadth First Search (BFS)** is started at node `2`. The nodes are listed in the order they are first visited. Which of the following is (are) possible output(s)?

Options :

6406531153167. ✓ `2 6 3 4 1 5`

6406531153168. ✓ `2 3 6 1 4 5`

6406531153169. ✗ `2 6 4 5 1 3`

6406531153170. ✓ `2 3 6 4 1 5`

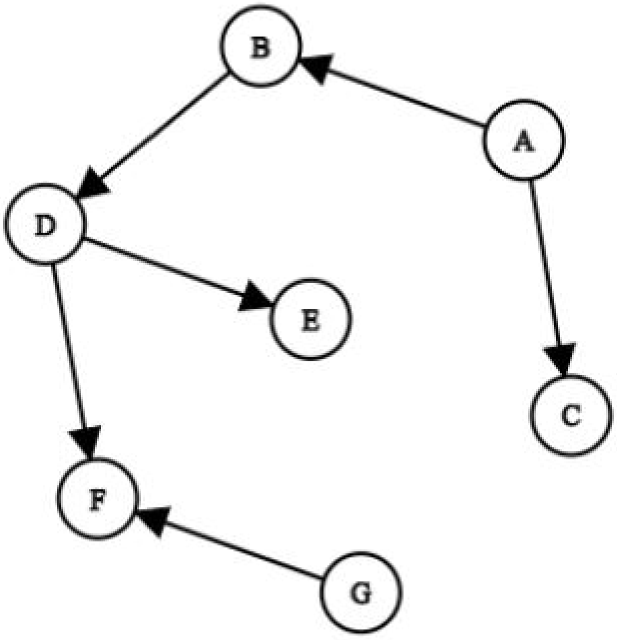
6406531153171. ✗ `2 6 1 3 4 5`

Question Number : 103 Question Id : 640653346898 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Select Question

Consider a DAG with V = {A, B, C, D, E, F, G}, shown below. Which of the following is/are **valid** topological orderings of the DAG?



Options :

6406531153172. ✖ A B C D E F G

6406531153173. ✔ A B G C D E F

6406531153174. ✖ G A C F B D E

6406531153175. ✔ A G C B D F E

6406531153176. ✔ G A B D F C E

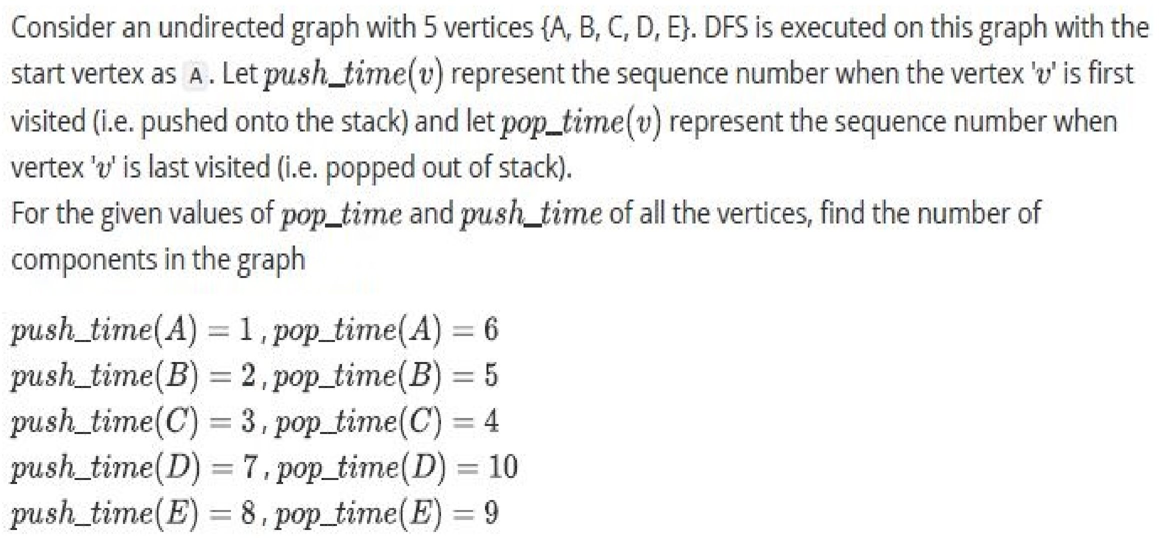
Sub-Section Number :	6
Sub-Section Id :	64065349208
Question Shuffling Allowed :	Yes

Question Number : 104 Question Id : 640653346899 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Short Answer Question



Consider an undirected graph with 5 vertices {A, B, C, D, E}. DFS is executed on this graph with the start vertex as `A`. Let $push\_time(v)$ represent the sequence number when the vertex 'v' is first visited (i.e. pushed onto the stack) and let $pop\_time(v)$ represent the sequence number when vertex 'v' is last visited (i.e. popped out of stack).

For the given values of pop_time and $push\_time$ of all the vertices, find the number of components in the graph

$$push\_time(A) = 1\,, pop\_time(A) = 6$$
$$push\_time(B) = 2\,, pop\_time(B) = 5$$
$$push\_time(C) = 3\,, pop\_time(C) = 4$$
$$push\_time(D) = 7\,, pop\_time(D) = 10$$
$$push\_time(E) = 8\,, pop\_time(E) = 9$$

NOTE: Enter your answer to the nearest integer.

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

2

AppDev-1

Section Id :	64065321950
Section Number :	7
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and	Yes