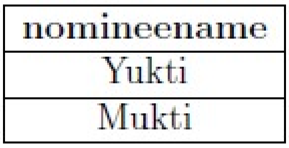
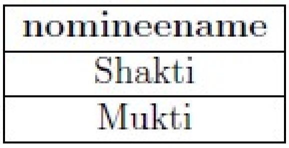
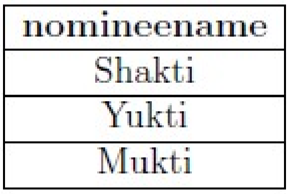
6406532239742. ✖ 

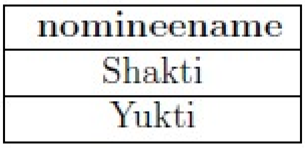
| nomineename |
|---|
| Yukti |
| Mukti |

6406532239743. ✖ 

| nomineename |
|---|
| Shakti |
| Mukti |

6406532239744. ✖ 

| nomineename |
|---|
| Shakti |
| Yukti |
| Mukti |

6406532239745. ✔ 

| nomineename |
|---|
| Shakti |
| Yukti |

# PDSA

| | |
|---|---|
| **Section Id :** | 64065344900 |
| **Section Number :** | 6 |
| **Section type :** | Online |
| **Mandatory or Optional :** | Mandatory |
| **Number of Questions :** | 16 |
| **Number of Questions to be attempted :** | 16 |
| **Section Marks :** | 50 |
| **Display Number Panel :** | Yes |
| **Section Negative Marks :** | 0 |
| **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 :** | 64065395155 |
| **Question Shuffling Allowed :** | No |
| **Is Section Default? :** | null |

**Question Number : 85 Question Id : 640653668525 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 "DIPLOMA LEVEL : PROGRAMMING, DATA STRUCTURES AND ALGORITHMS USING PYTHON (COMPUTER BASED EXAM)"**

**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 :**

6406532239754. ✓ YES

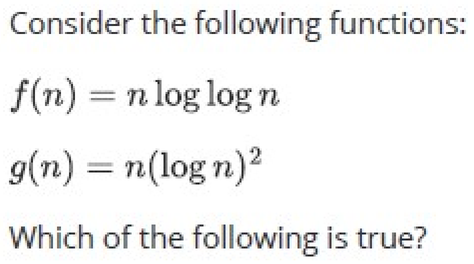
6406532239755. ✗ NO

| | |
|---|---|
| **Sub-Section Number :** | 2 |
| **Sub-Section Id :** | 64065395156 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 86 Question Id : 640653668526 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

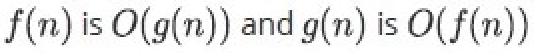

Consider the following functions:

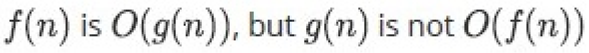
$f(n) = n \log \log n$

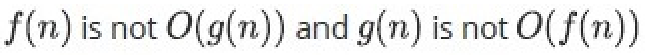
$g(n) = n(\log n)^2$

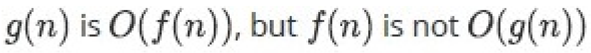
Which of the following is true?

**Options :**

6406532239756. ✖ 
$f(n)$ is $O(g(n))$ and $g(n)$ is $O(f(n))$

6406532239757. ✓ 
$f(n)$ is $O(g(n))$, but $g(n)$ is not $O(f(n))$

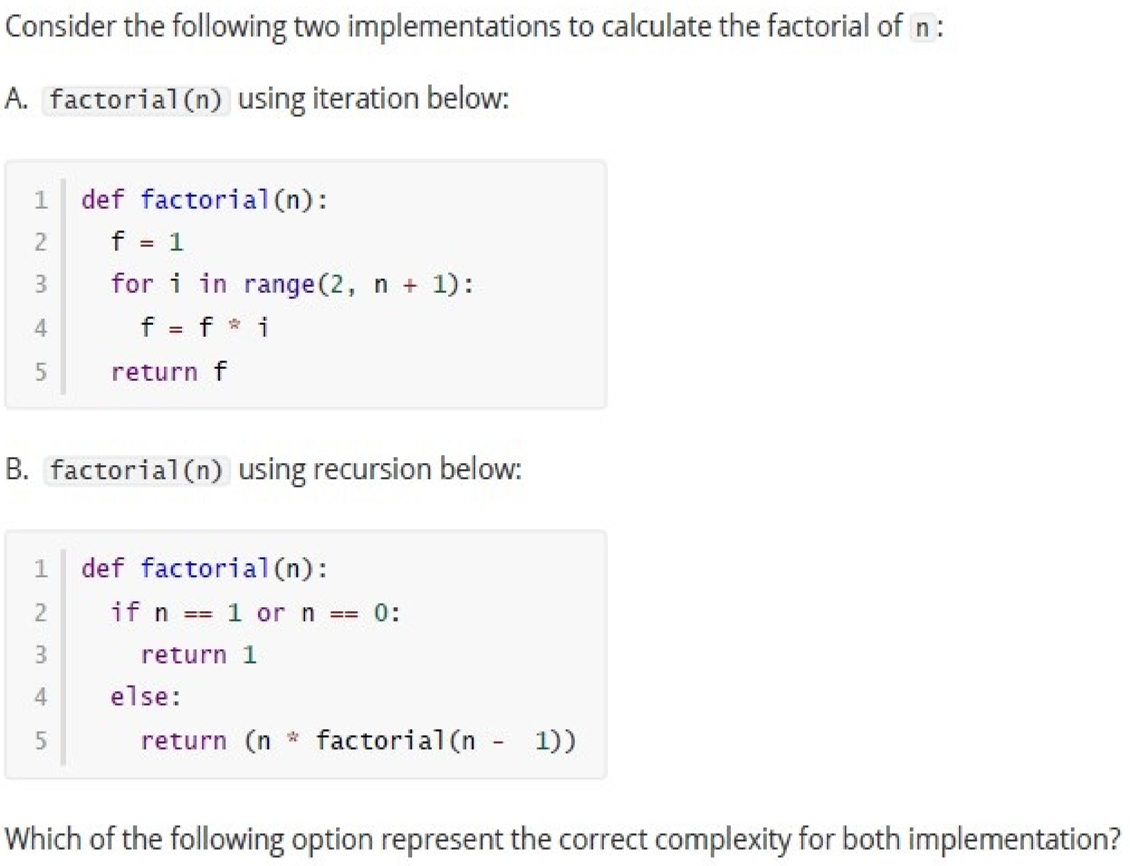
6406532239758. ✖ 
$f(n)$ is not $O(g(n))$ and $g(n)$ is not $O(f(n))$

6406532239759. ✖ 
$g(n)$ is $O(f(n))$, but $f(n)$ is not $O(g(n))$

**Question Number : 87 Question Id : 640653668527 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 two implementations to calculate the factorial of `n`:

A. `factorial(n)` using iteration below:

```
def factorial(n):
  f = 1
  for i in range(2, n + 1):
    f = f * i
  return f
```

B. `factorial(n)` using recursion below:

```
def factorial(n):
  if n == 1 or n == 0:
    return 1
  else:
    return (n * factorial(n -  1))
```

Which of the following option represent the correct complexity for both implementation?

**Options :**

6406532239760. ✗ A - $O(n)$, B - $O(n^2)$

6406532239761. ✗ A - $O(n)$, B - $O(log\ n)$

6406532239762. ✗ A - $O(n^2)$, B - $O(n^2)$

6406532239763. ✓ A - $O(n)$, B - $O(n)$

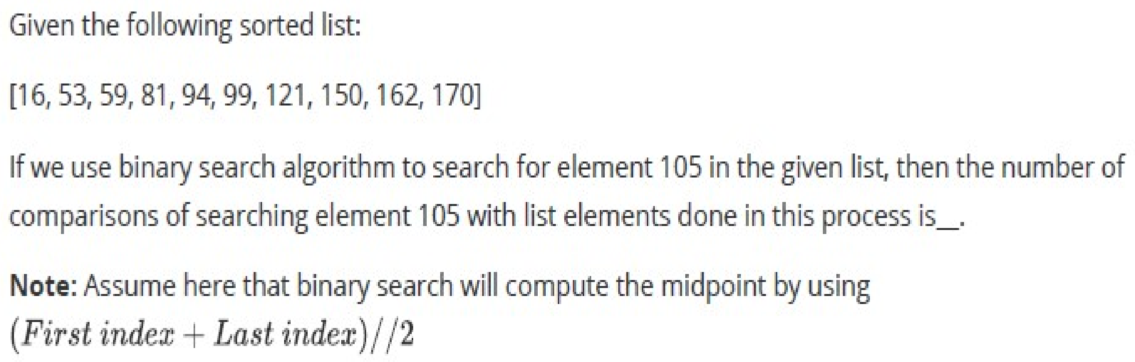
| | |
|---|---|
| **Sub-Section Number :** | 3 |
| **Sub-Section Id :** | 64065395157 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 88 Question Id : 640653668528 Question Type : SA Calculator : None**

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

**Correct Marks : 3**

Question Label : Short Answer Question



Given the following sorted list:

[16, 53, 59, 81, 94, 99, 121, 150, 162, 170]

If we use binary search algorithm to search for element 105 in the given list, then the number of comparisons of searching element 105 with list elements done in this process is__.

**Note:** Assume here that binary search will compute the midpoint by using $(First\ index + Last\ index)//2$

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Equal

**Text Areas :** PlainText

**Possible Answers :**

4

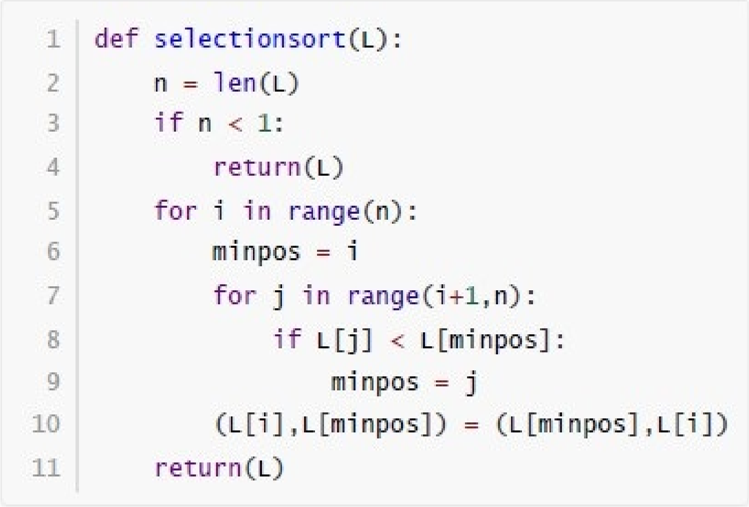
| | |
|---|---|
| **Sub-Section Number :** | 4 |
| **Sub-Section Id :** | 64065395158 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 89 Question Id : 640653668529 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 3 Max. Selectable Options : 0**

Question Label : Multiple Select Question

Consider the following **Selection sort** algorithm :



```
def selectionsort(L):
    n = len(L)
    if n < 1:
        return(L)
    for i in range(n):
        minpos = i
        for j in range(i+1,n):
            if L[j] < L[minpos]:
                minpos = j
        (L[i],L[minpos]) = (L[minpos],L[i])
    return(L)
```

Which of the following statement(s) is/are true about  the given Selection sort algorithm?

**Options :**

6406532239765. ✓ It is considered an unstable sorting algorithm.

6406532239766. ✗ It is considered a stable sorting algorithm.

6406532239767. ✓ It has the same time complexity irrespective of the sequence of elements in the input.

6406532239768. ✗ It is efficient for larger data sets.

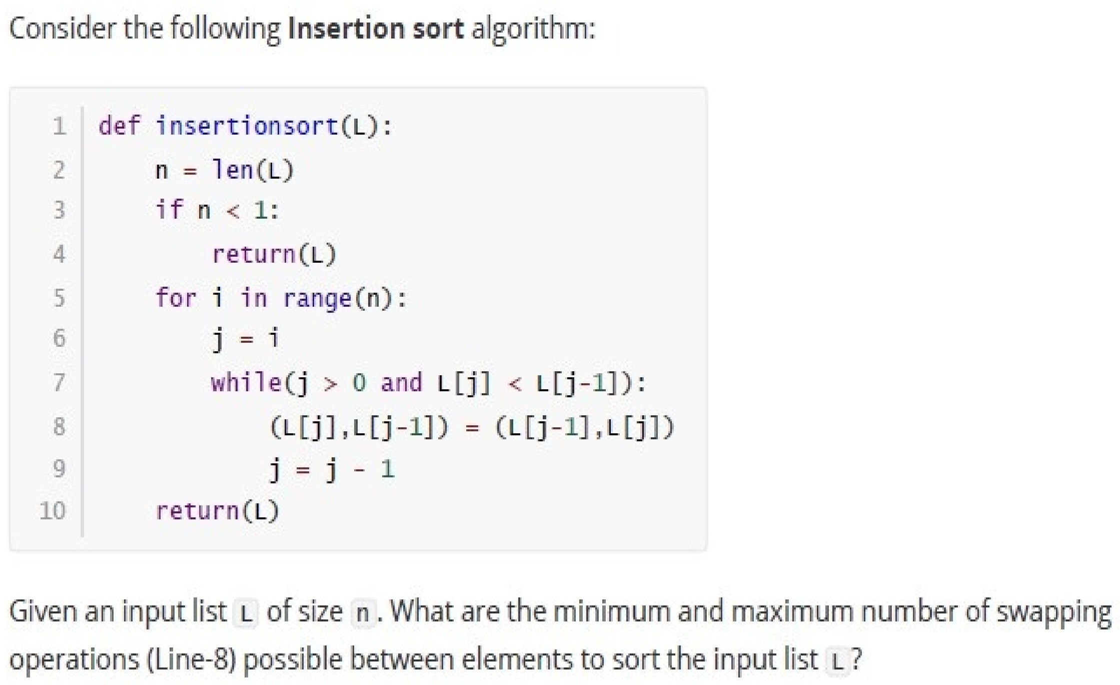
6406532239769. ✓ It is sorted in place.

|  |  |
|---|---|
| **Sub-Section Number :** | 5 |
| **Sub-Section Id :** | 64065395159 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 90 Question Id : 640653668530 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 **Insertion sort** algorithm:

```
1   def insertionsort(L):
2       n = len(L)
3       if n < 1:
4           return(L)
5       for i in range(n):
6           j = i
7           while(j > 0 and L[j] < L[j-1]):
8               (L[j],L[j-1]) = (L[j-1],L[j])
9               j = j - 1
10      return(L)
```

Given an input list `L` of size `n`. What are the minimum and maximum number of swapping operations (Line-8) possible between elements to sort the input list `L`?

**Options :**

6406532239770. ✖ Minimum: 0, Maximum: $n(n+1)/2$

6406532239771. ✖ Minimum: $n-1$, Maximum: $n(n+1)/2$

6406532239772. ✓ Minimum: 0, Maximum: $n(n-1)/2$

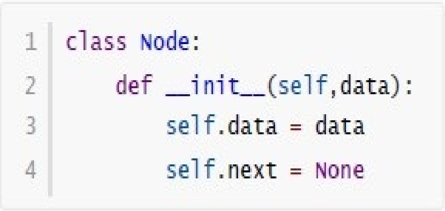
6406532239773. ✖ Minimum: $n-1$, Maximum: $n(n-1)/2$

6406532239774. ✖ Minimum: 0, Maximum: $n^2$

**Question Number : 91 Question Id : 640653668532 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

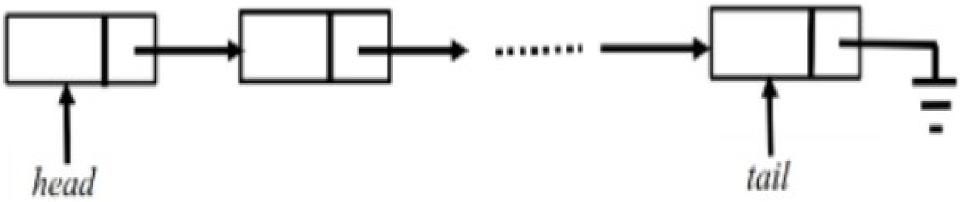
**Correct Marks : 3**

Question Label : Multiple Choice Question



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

Consider the following linked list structure, where each node is an object of class `Node` and it has a `head` pointer that points to the first node of the linked list and a `tail` pointer that points to the last node of the linked list.



Which of the following operations on a given linked list requires traversal of the entire list?

**Options :**

6406532239779. ✗ Insert a new node at the beginning

6406532239780. ✓ Delete a node from the end

6406532239781. ✗ Insert a new node at the end

6406532239782. ✗ Delete a node from the beginning

**Question Number : 92 Question Id : 640653668533 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

A doctor sees patients in his consulting room every evening. A certain number of appointments are given for each evening. Patients can also take a chance and come without an appointment. Appointments are not for a fixed time. The receptionist allows patients in to see the doctor in the order in which they arrive, with the provision that any patient with an appointment goes in before any patient without an appointment. What would be a good data structure for the receptionist to keep track of the waiting patients?

**Options :**

6406532239783.

✖ Single stack

6406532239784. ✖ Two stacks

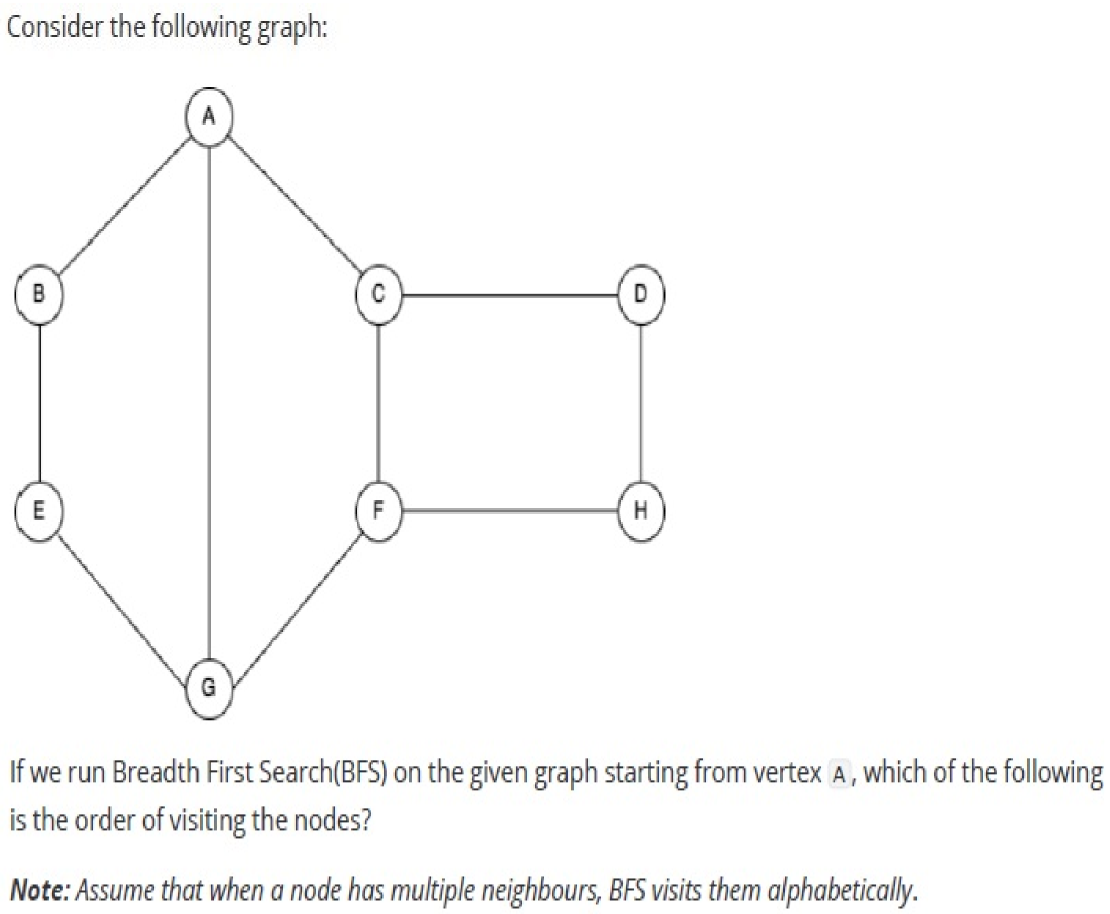
6406532239785. ✖ Single queue

6406532239786. ✔ Two queues

**Question Number : 93 Question Id : 640653668537 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 graph:

If we run Breadth First Search(BFS) on the given graph starting from vertex A, which of the following is the order of visiting the nodes?

***Note:*** *Assume that when a node has multiple neighbours, BFS visits them alphabetically.*

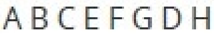
**Options :**

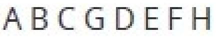
6406532239793. ✖

A B C G E F D H

6406532239794. ✔

A B C G E D F H

6406532239795. ✖ 
A B C E F G D H

6406532239796. ✖ 
A B C G D E F H

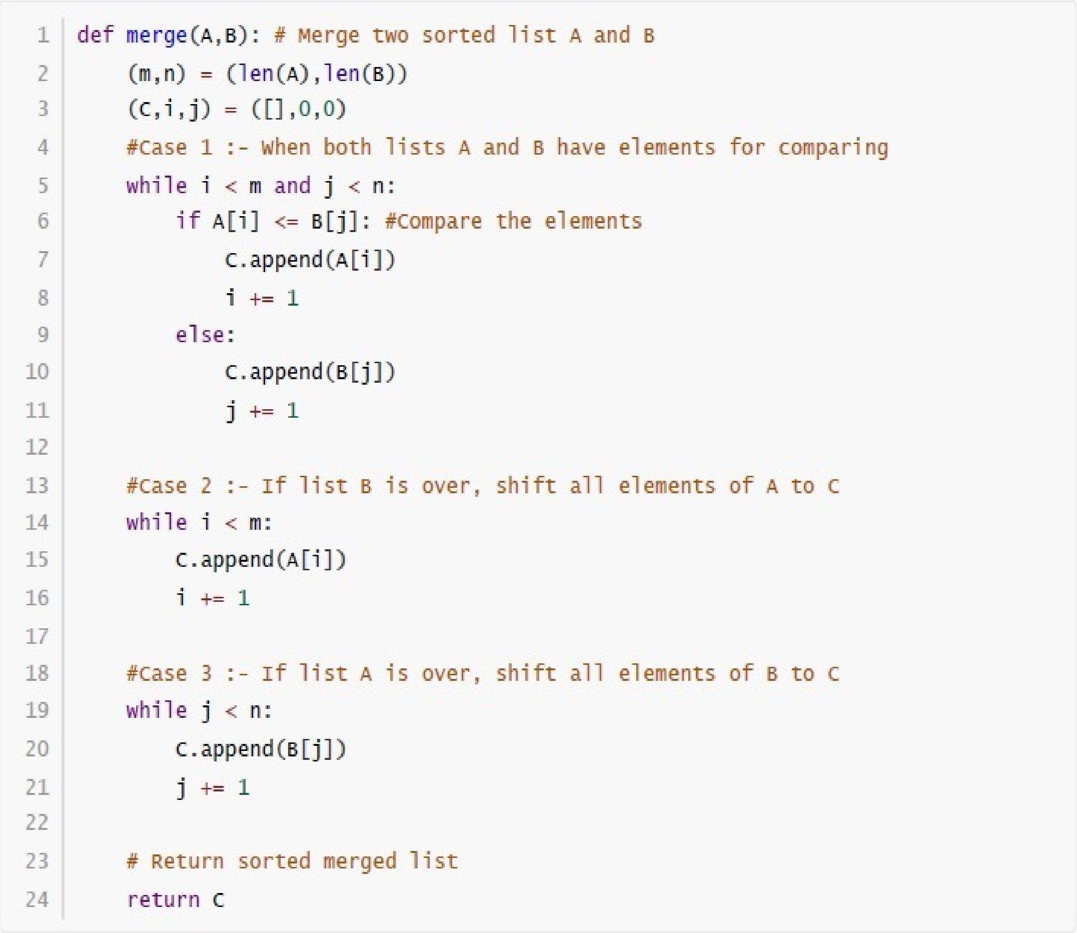
| | |
|---|---|
| **Sub-Section Number :** | 6 |
| **Sub-Section Id :** | 64065395160 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 94 Question Id : 640653668531 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

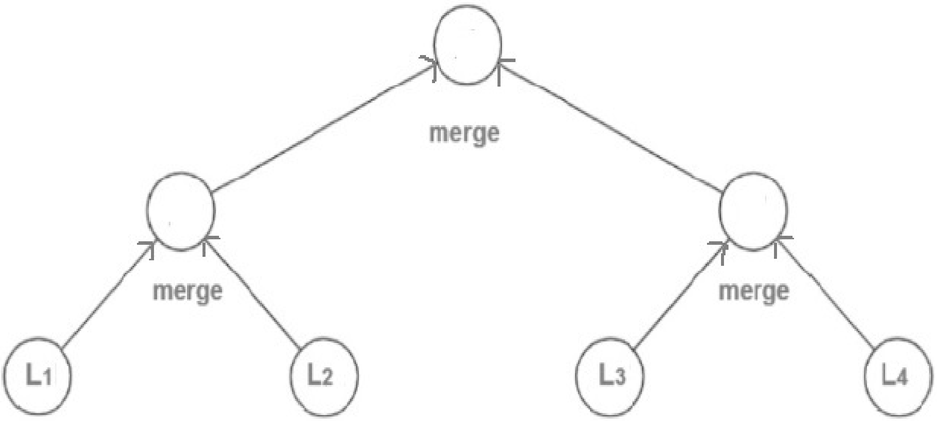
Consider the following implementation for merge two sorted list and return one sorted merged list:



```
def merge(A,B): # Merge two sorted list A and B
    (m,n) = (len(A),len(B))
    (C,i,j) = ([],0,0)
    #Case 1 :- When both lists A and B have elements for comparing
    while i < m and j < n:
        if A[i] <= B[j]: #Compare the elements
            C.append(A[i])
            i += 1
        else:
            C.append(B[j])
            j += 1

    #Case 2 :- If list B is over, shift all elements of A to C
    while i < m:
        C.append(A[i])
        i += 1

    #Case 3 :- If list A is over, shift all elements of B to C
    while j < n:
        C.append(B[j])
        j += 1

    # Return sorted merged list
    return C
```

If 4 sorted lists $L_1$, $L_2$, $L_3$ and $L_4$ each of length $n/2$ are merged into a single sorted list of elements using two-way merging by given `merge` function. What will be the minimum and maximum number of element comparisons needed for this process?

**Two-way merge:**



**Options :**

6406532239775. ✓ Minimum: $2n$, Maximum: $4n-3$

6406532239776. ✖ Minimum: $2n$, Maximum: $4n-1$

6406532239777.

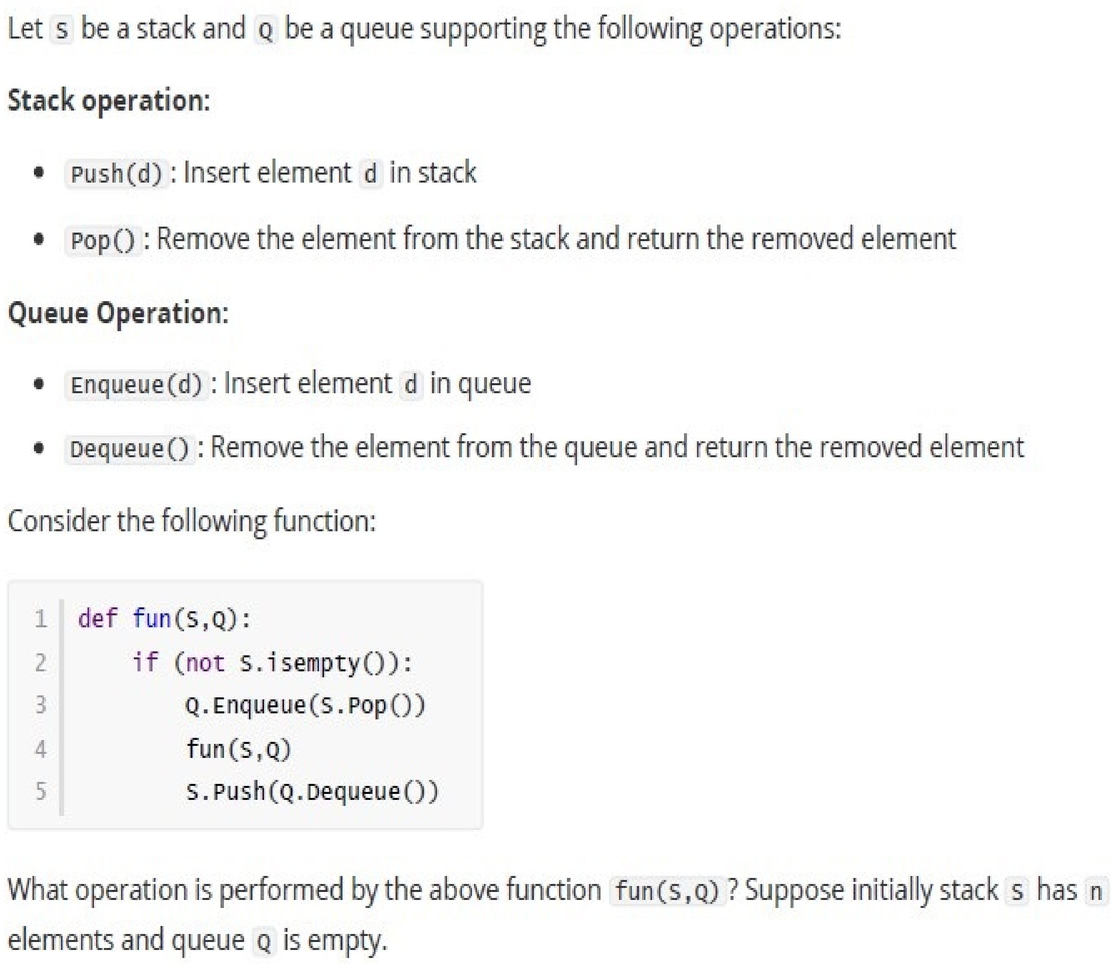
✖ Minimum: $2n-1$, Maximum: $4n-3$

6406532239778. ✖ Minimum: $2n-1$, Maximum: $4n+1$

**Question Number : 95 Question Id : 640653668534 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



Let `S` be a stack and `Q` be a queue supporting the following operations:

**Stack operation:**

- `Push(d)` : Insert element `d` in stack
- `Pop()` : Remove the element from the stack and return the removed element

**Queue Operation:**

- `Enqueue(d)` : Insert element `d` in queue
- `Dequeue()` : Remove the element from the queue and return the removed element

Consider the following function:

```
def fun(S,Q):
    if (not S.isempty()):
        Q.Enqueue(S.Pop())
        fun(S,Q)
        S.Push(Q.Dequeue())
```

What operation is performed by the above function `fun(S,Q)` ? Suppose initially stack `S` has `n` elements and queue `Q` is empty.

**Options :**

6406532239787. ✖ Leaves the stack S unchanged

6406532239788. ✔ Reverses the order of the elements in the stack S

6406532239789. ✖ Swap the top and bottom element of the stack S, keeping the other elements in the same order

6406532239790. ✖ Empties the stack S

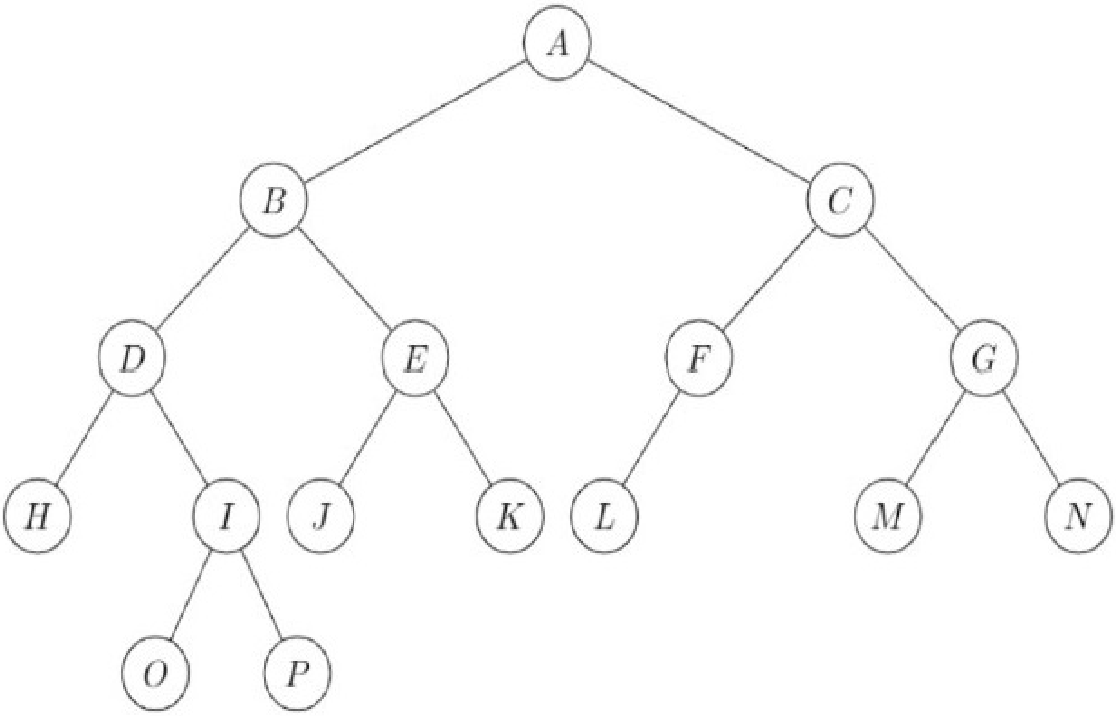
| | |
|---|---|
| **Sub-Section Number :** | 7 |
| **Sub-Section Id :** | 64065395161 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 96 Question Id : 640653668538 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 4 Max. Selectable Options : 0**

Question Label : Multiple Select Question

Suppose we obtain the following DFS tree rooted at node A for an undirected graph with vertices {A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P}.



Which of the following cannot be an edge/edges in the original graph?

**Options :**

6406532239797. ✖ (A, K)

6406532239798.

✓ (H, J)

6406532239799. ✓ (M, N)

6406532239800. ✖ (B, O)

6406532239801. ✓ (P, E)

6406532239802. ✖ (C, M)

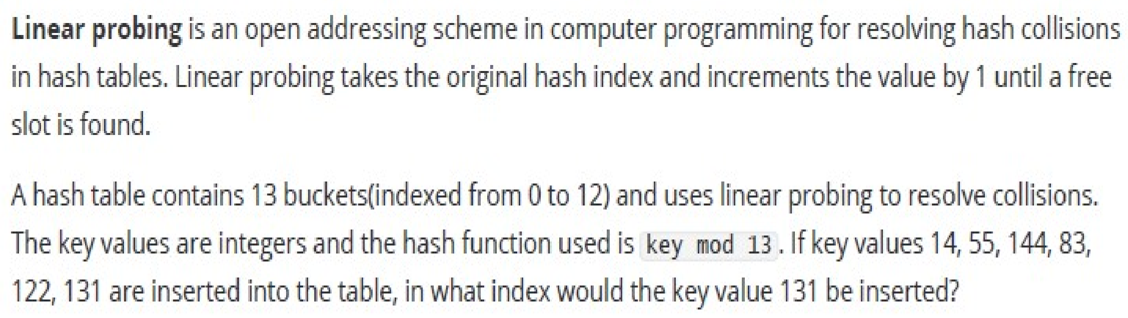
| | |
|---|---|
| **Sub-Section Number :** | 8 |
| **Sub-Section Id :** | 64065395162 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 97 Question Id : 640653668535 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



**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.

A hash table contains 13 buckets(indexed from 0 to 12) and uses linear probing to resolve collisions. The key values are integers and the hash function used is `key mod 13`. If key values 14, 55, 144, 83, 122, 131 are inserted into the table, in what index would the key value 131 be inserted?

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Equal

**Text Areas :** PlainText

**Possible Answers :**

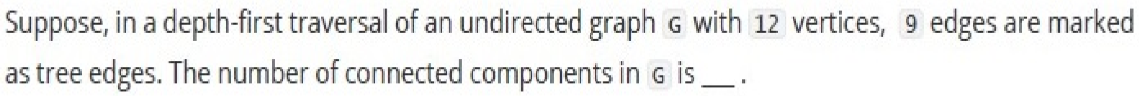
4

**Question Number : 98 Question Id : 640653668539 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



Suppose, in a depth-first traversal of an undirected graph `G` with `12` vertices, `9` edges are marked as tree edges. The number of connected components in `G` is ___ .

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Equal

**Text Areas :** PlainText

**Possible Answers :**

3

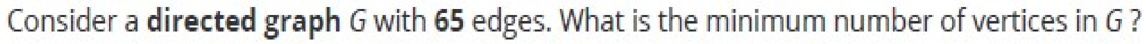
| | |
|---|---|
| **Sub-Section Number :** | 9 |
| **Sub-Section Id :** | 64065395163 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 99 Question Id : 640653668536 Question Type : SA Calculator : None**

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

**Correct Marks : 3**

Question Label : Short Answer Question



Consider a **directed graph** *G* with **65** edges. What is the minimum number of vertices in *G* ?

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Equal

**Text Areas :** PlainText

**Possible Answers :**

9

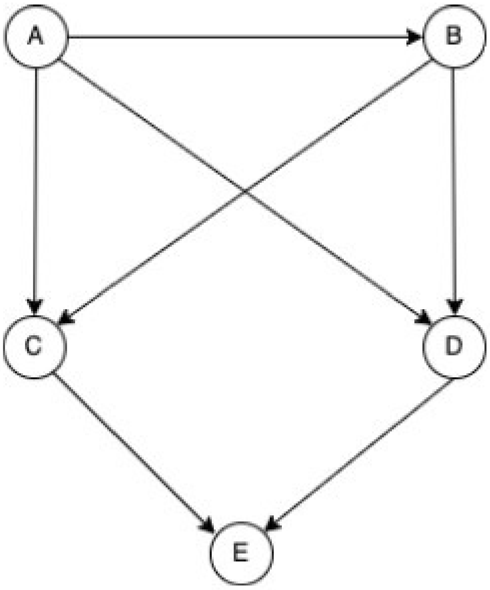
**Question Number : 100 Question Id : 640653668540 Question Type : SA Calculator : None**

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

**Correct Marks : 3**

Question Label : Short Answer Question

Consider the following Directed Acyclic Graph(DAG):



The number of possible topological order(s) for the given graph is_____________.

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Equal

**Text Areas :** PlainText

**Possible Answers :**

2

# AppDev1

| | |
|---|---|
| **Section Id :** | 64065344901 |
| **Section Number :** | 7 |
| **Section type :** | Online |