

The due date for submitting this assignment has passed.

Due on 2022-05-15, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

Note: This assignment will be evaluated after the deadline passes. You will get your score 48 hrs after the deadline. Until then the score will be shown as Zero.

```
1) 1 def fun(s):
2     p = 0
3     s = s.lower()
4     for i in range(len(s)):
5         if s[i] not in s[:i]:
6             p += 1
7     return p
```

1 point

S is a non-empty string of English letters without any space. What fun(S) will return after execution of the above code?

- Total number of letters in the string S.
- Total number of distinct letters in the string S.
- Total number of letters that are repeated in the string S more than one time.
- Difference of total letters in the string S and distinct letters in the string S.

No, the answer is incorrect.

Score: 0

Feedback:

During traversing of the input string , p is incremented by one only when any letter occurs first time. Hence, the return value of p represents the total number of distinct letters in the string S.

Accepted Answers:

Total number of distinct letters in the string S .

2) Which of the following is/are valid reason(s) for NameError exception? [MSQ]

1 point

- Variable is not defined.
- Calling a function before declaration.
- Misspelled built-in functions name
- Variables are defined globally in the program .

No, the answer is incorrect.

Score: 0

Feedback:

Options (a), (b) and (c) are valid reasons for 'NameError' exception. But option (d) is not correct because the global variable can be accessed from any scope in the program, so it will not return any error.

Accepted Answers:

Variable is not defined.

Calling a function before declaration.

Misspelled built-in functions name

```
3) 1 def f(n):
2     s = 0
3     for i in range(2,n):
4         if n % i == 0 and i % 2 == 1:
5             s = s + 1
6     return(s)
```

What is f(60) - f(59) , given the definition of f above?

No, the answer is incorrect.

Score: 0

Accepted Answers:

(Type: Numeric) 3

1 point

```
4) 1 x = 1
2 while True:
3     if x % 5 == 0:
4         break
5     print(x, end = ' ')
6     x += 1
```

1 point

What will be the output of the above code-snippet?

- Syntax error
- 2 1
- 0 3 1
- None of these

No, the answer is incorrect.

Score: 0

Feedback:

In given code-snippet, in line 3, comparison operator `==` has one extra space in between, because of that, the code will raise a 'Syntax error'.

Accepted Answers:

Syntax error

```
5) 1 class Person:
2     def __init__(self, name):
3         self.name = name
4     def say_hi(self):
5         print('Hello, ', self.name)
6
7 p = Person('Good morning')
8 p.say_hi()
```

1 point

What will be the output of the above code-snippet?

- Good morning
- Hello, Good morning
- Hello Good morning
- Good

No, the answer is incorrect.

Score: 0

Feedback:

List 'a' has three elements (index 0 to 2). Inside 'try' block first print statement will execute correctly after that second print statement will raise an index error because index 3 is not available in list 'a'. Due to this error 'except' block's print statement will execute. Hence, option (c) is correct.

Accepted Answers:

L = [4, 2, 4]
L = [8, 4, 16]
L = [48, 6, 36]

No, the answer is incorrect.

Score: 0

Feedback:

The 'ZeroDivisionError' is due to if a number being divided by zero. In code line number 4, if divisor `(L[1]**2 - L[2])` becomes zero, then it produces a 'ZeroDivisionError' exception. For options (b), (c) and (d), divisor `(L[1]**2 - L[2])` become zero. Hence, For options (b), (c) and (d) are correct.

Accepted Answers:

L = [4, 2, 4]
L = [8, 4, 16]
L = [48, 6, 36]

```
6) 1 a = [1, 2, 3]
2 try:
3     print ("Second element = %d" %(a[1]))
4     print ("Fourth element = %d" %(a[3]))
5 except:
6     print ("An error occurred")
```

1 point

What will be the output of the above code-snippet?

- 1 Second element = 2
- An error occurred
- 1 Second element = 2
2 An error occurred
- 1 Second element = 2
2 Fourth element = 3
3 An error occurred

No, the answer is incorrect.

Score: 0

Feedback:

An 'IndexError' occurs if the code is trying to access an index that does not exist. The 'pop(i)' function removes the 'i'th indexed element from the list and returns the removed element. Line number 4 compares the first and last indexed elements of the list after removing them from the list. If input list length is odd and the first removed and last removed element are equal, then for every cycle of while loop, the first and last element will be removed. In the last cycle of the while loop, only one element will remain in the list, So 'pop(0)' will remove this single element from the list then 'pop(-1)' will produce an index error because the list is already empty. Hence, options (a), (c) and (d) are correct.

Accepted Answers:

L = [1, 2, 3, 4, 3, 2, 1]
L = [2, 2, 2, 2, 2]
L = [1, 1, 1, 1, 1, 1]
L = [8]

No, the answer is incorrect.

Score: 0

Feedback:

In the given code, 'name', 'course' are object variables and 'count' is a class variable and the count is incremented by one when any new object is created. Hence, options (c) and (e) are correct.

Accepted Answers:

name and course are object variables, and count is a class variable.
count represents the number of objects created for class Enrollment

```
7) 1 def special3Bad(L):
2     try:
3         if L[0] % L[1] == 0 and L[1] != 0:
4             if L[0] / (L[1]**2 - L[2]) == 0:
5                 return True
6             return False
7     except ZeroDivisionError:
8         print('zerodivisionerror')
9     except:
10        print('some other exception occurred')
11    else:
12        print('No exception occurred')
13    special3Bad(L)
```

1 point

Given above is a function that checks whether a list satisfies some property. There is an error in this function. Select the list(s) L = [n1, n2, ..., n3], where n1, n2 and n3 are all integers, for which special3Bad(L) produces a ZeroDivisionError exception. [MSQ]

Accepted Answers:

L = [4, 2, 8]
L = [4, 2, 4]
L = [8, 4, 16]
L = [48, 6, 36]

No, the answer is incorrect.

Score: 0

Feedback:

The 'ZeroDivisionError' is due to if a number being divided by zero. In code line number 4, if divisor `(L[1]**2 - L[2])` becomes zero, then it produces a 'ZeroDivisionError' exception. For options (b), (c) and (d), divisor `(L[1]**2 - L[2])` become zero. Hence, For options (b), (c) and (d) are correct.

Accepted Answers:

L = [4, 2, 4]
L = [8, 4, 16]
L = [48, 6, 36]

```
8) 1 def isSymmetricBad(L):
2     try:
3         while len(L) > 0:
4             if L.pop(0) != L.pop(-1):
5                 return False
6             return True
7     except IndexError:
8         print('IndexError')
9     except:
10        print('Some other exception occurred')
11    else:
12        print('No exception occurred')
13    isSymmetricBad(L)
```

1 point

Given above is a function to check whether a list is a palindrome. There is an error in this function. Select the list(s) L = [n1, n2, ..., n2, n1], for which isSymmetricBad(L) produces an IndexError exception. [MSQ]

Accepted Answers:

L = [1, 2, 3, 4, 3, 2, 1]
L = [2, 2, 2, 2, 2]
L = [1, 1, 1, 1, 1, 1]
L = [8]

No, the answer is incorrect.

Score: 0

Feedback:

An 'IndexError' occurs if the code is trying to access an index that does not exist. The 'pop(i)' function removes the 'i'th indexed element from the list and returns the removed element. Line number 4 compares the first and last indexed elements of the list after removing them from the list. If input list length is odd and the first removed and last removed element are equal, then for every cycle of while loop, the first and last element will be removed. In the last cycle of the while loop, only one element will remain in the list, So 'pop(0)' will remove this single element from the list then 'pop(-1)' will produce an index error because the list is already empty. Hence, options (a), (c) and (d) are correct.

Accepted Answers:

L = [1, 2, 3, 4, 3, 2, 1]
L = [2, 2, 2, 2, 2]
L = [1, 1, 1, 1, 1, 1]
L = [8]

```
9) 1 def gcd(m,n):
2     (a,b) = (max(m,n), min(m,n))
3     if a % b == 0:
4         return b
5     else:
6         return gcd(b, a % b)
7 print(gcd(24,130))
```

1 point

How many times gcd() function will be called?

Note: Ignore the first call given in the code.

No, the answer is incorrect.

Score: 0

Feedback:

In the given code, 'name', 'course' are object variables and 'count' is a class variable. For options (c) and (e) are correct.

Accepted Answers:

name and course are object variables, and count is a class variable.
count represents the number of objects created for class Enrollment

```
10) 1 class Enrollment:
2     count = 0
3     def __init__(self, n, c):
4         self.name = n
5         self.course = c
6         Enrollment.count += 1
7     def display(self):
8         print(self.name)
9         print(self.course)
```

1 point

Which of the following option(s) is/are correct about the given code? [MSQ]

- count , name and course are object variables.
- name and course are class variable and count is an object variable.
- name and course are object variables, and count is a class variable.
- count , name and course are class variables.
- count represents the number of objects created for class Enrollment

No, the answer is incorrect.

Score: 0

Feedback:

In the given code, 'name', 'course' are object variables and 'count' is a class variable and the count is incremented by one when any new object is created. Hence, options (c) and (e) are correct.

Accepted Answers:

name and course are object variables, and count is a class variable.
count represents the number of objects created for class Enrollment

The due date for submitting this assignment has passed.

Due on 2022-05-15, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

Note: This assignment will be evaluated after the deadline passes. You will get your score 48 hrs after the deadline. Until then the score will be shown as Zero.

1)

```
1 def fun(n):
2     s = 0
3     for i in range(0, n):
4         for j in range(0, n):
5             for k in range(0, n):
6                 s += 1
7     for i in range(0, n):
8         for j in range(0, n):
9             s -= 1
10    for i in range(0, n):
11        s -= 1
12    return s
```

1 point

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

- $O(n^6)$
- $O(n^2)$
- $O(n^3)$
- $O(n^5)$

No, the answer is incorrect.

Score: 0

Feedback:

First nested for loop runs $O(n^3)$ then second nested for loop runs in $O(n^2)$ then the third loop runs $O(n)$. That is why time complexity will be $O(n^3)$ by taking higher-order terms.

Accepted Answers:

$O(n^3)$

2) Let $B(n)$, $A(n)$ and $W(n)$ be best-case, average-case, and worst-case running time of an algorithm, executed on an input size n . Which of the following is/are always True.

1 point

- $A(n) = O(B(n))$
- $A(n) = O(W(n))$
- $A(n) = \Omega(W(n))$
- $A(n) = \Omega(B(n))$
- $B(n) = O(W(n))$
- $B(n) = O(A(n))$

No, the answer is incorrect.

Score: 0

Feedback:

Big O notation gives upper bound and Ω notation gives lower bound, then ' $B(n) \leq A(n) \leq W(n)$ ' condition should be satisfied. Hence, (a) and (c) can not be the correct option.

Accepted Answers:

$A(n) = O(W(n))$
 $A(n) = \Omega(B(n))$
 $B(n) = O(W(n))$
 $B(n) = O(A(n))$

3) What is the asymptotic complexity of merge sort when the input is sorted in reverse order?

1 point

- $O(n \log n^2)$
- $O(n^2 \log n)$
- $O(n \log n)$
- $O(n^2)$

No, the answer is incorrect.

Score: 0

Feedback:

Since Merge sort always divides the array into two halves and takes linear time to merge two halves. That is why in every case (Best Average, Worst) time complexity is the same $O(n \log n)$

Accepted Answers:

$O(n \log n)$

4)

```
1 def selectionsort(L):
2     n = len(L)
3     if n < 1:
4         return(L)
5     for i in range(n):
6         mpos = i
7         for j in range(i + 1, n):
8             if L[j] < L[mpos]:
9                 mpos = j
10            L[i], L[mpos] = (L[mpos], L[i])
11    return(L)
```

1 point

What is the value of i when the list [9, 4, 5, 2, 3, 7, 6, 8, 1] becomes completely sorted for the first time?



No, the answer is incorrect.

Score: 0

Accepted Answers:

(Type: Numeric) 5

1 point

5)

```
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)
```

1 point

Which of the following statement(s) is/are correct with regard to the given insertion sort? [MSQ]

- The sort is stable and it does not sort in-place
- The sort is unstable and it sorts in-place
- The sort is stable and it sorts in-place
- After m iterations of the for-loop, the first m elements in the list are in sorted order
- After m iterations of the for-loop, the first m elements in the list are the m smallest elements of the list

No, the answer is incorrect.

Score: 0

Feedback:

Insertion sort is stable and in-place since it preserves the original element order for equal value and does not require extra space. We can clearly see that after m iteration, it is sorting only the first m element.

Accepted Answers:

The sort is stable and it sorts in-place
After m iterations of the for-loop, the first m elements in the list are in sorted order

6) A program is written in 3 stages where the first stage is of $O(n \log n)$, the second stage is of $O(n^2)$ which is based on the result of the first stage, and the third stage is of $O(n)$. What will be asymptotic complexity of the entire program?

1 point

- $O(n)$
- $O(n^2)$
- $O(n^3)$
- $O(n^2 \log n)$

No, the answer is incorrect.

Score: 0

Feedback:

$f_3(n) = 3n + \log n$ we can find such c . if $x > y$ then we can't find constant c such that $n^x \leq cn^y$ and if $x < y$ we can't find constant c such that $n^x \geq cn^y$ hence, it is true only if $x = y$ and vice versa.

Accepted Answers:

$f_3(n) = 3n + \log n$

7) A school wants to maintain a database of its students. Each student has a unique id and it is stored along with other details. Adding a new student with a unique id, searching for a student using their id, removing of students are the frequent operations performed on the database. From the options given below, choose the most efficient technique to store the data.

1 point

- Maintain a sorted list with id. Whenever a new student is added, append the student details at the end, and sort the entire list using selection sort.
- Maintain a sorted list with id. Whenever a new student is added, append the student details at the end, and sort the entire list using insertion sort.
- Maintain a sorted list with id. Whenever a new student is added, append the student details at the end, and sort the entire list using merge sort.
- Maintain a sorted list with id. Whenever a new student is added, insert the student details into the respective position in the sorted list by id.

No, the answer is incorrect.

Score: 0

Feedback:

Insertion sort concept is best suited for the given scenario. That is why option (d) is correct.

Accepted Answers:

Maintain a sorted list with id. Whenever a new student is added, insert the student details into the respective position in the sorted list by id.

8)

```
1 def tsearch(L, x):
2     global c
3     c += 1
4     n = len(L)
5
6     if n == 0:
7         return False
8     if L[n // 3] == x:
9         return True
10    if L[2 * n // 3] == x:
11        return True
12
13    if x < L[n // 3]:
14        return tsearch(L[:n // 3], x)
15    elif x > L[2 * n // 3]:
16        return tsearch(L[2 * n // 3:], x)
17    else:
18        return tsearch(L[n // 3 : 2 * n // 3], x)
```

1 point

Choose the order of complexity of the search function **tsearch**.

- $O(\sqrt{n})$
- $O(\frac{\log n}{n^2})$
- $O(\log n)$
- $O(\frac{\log n}{n})$

No, the answer is incorrect.

Score: 0

Feedback:

$f_3(n) = 10^{31}$ we can verify this.

Accepted Answers:

$f_3(n) = 10^{31}$

9) Which of the following statement(s) is/are true?

1 point

I: $(n+3)^k = \Omega(n^k)$

II: $n^x = \Theta(n^y)$, if and only if $x = y$

- Only I is true.
- Only II is true.
- I and II both are true.
- I and II both are false.

No, the answer is incorrect.

Score: 0

Feedback:

$(n+3)^k \geq cn^k$ we can find such c . if $x > y$ then we can't find constant c such that $n^x \leq cn^y$ and if $x < y$ we can't find constant c such that $n^x \geq cn^y$ hence, it is true only if $x = y$ and vice versa.

Accepted Answers:

I and II both are true.

10) Arrange the following functions in increasing order of asymptotic complexity

1 point

$f_1(n) = 3n + \log n$

$f_2(n) = (\log n)^2$

$f_3(n) = \log(\log n)$

$f_4(n) = 100n \log n$

$f_5(n) = 3n \log n$

- $f_3(n), f_4(n), f_2(n), f_1(n), f_5(n)$
- $f_3(n), f_2(n), f_1(n), f_5(n), f_4(n)$
- $f_2(n), f_3(n), f_4(n), f_5(n), f_1(n)$
- $f_2(n), f_3(n), f_4(n), f_1(n), f_5(n)$

No, the answer is incorrect.

Score: 0

Feedback:

$f_3(n) = 10^{31}$ we can verify this.

Accepted Answers:

$f_3(n), f_4(n), f_2(n), f_1(n), f_5(n)$

The due date for submitting this assignment has passed.

Due on 2022-05-29, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

Note: This assignment will be evaluated after the deadline passes. You will get your score 48 hrs after the deadline. Until then the score will be shown as Zero.

1) For a directed acyclic graph G with n nodes and m edges, what is the asymptotic complexity of efficient algorithms for topological sorting of G , using adjacency matrix and adjacency list representations? 1 point

- $O(n + m), O(n + m)$
- $O(n + m), O(n^2)$
- $O(n^2), O(n + m)$
- $O(n^2), O(n^2)$

No, the answer is incorrect.

Score: 0

Feedback:

Using adjacency matrix we finds minimum indegree vertex in $O(n^2)$ and by using adjacency list we do it by BFS so we get it in $O(n + m)$

Accepted Answers:

$O(n^2), O(n + m)$

2) The total number of edges that a complete undirected graph with n vertices can have is _____ 1 point

[A complete graph is a simple undirected graph in which every pair of vertices is connected by a unique edge.]

- n^2
- $2n$
- $\frac{n(n+1)}{2}$
- $\frac{n(n-1)}{2}$

No, the answer is incorrect.

Score: 0

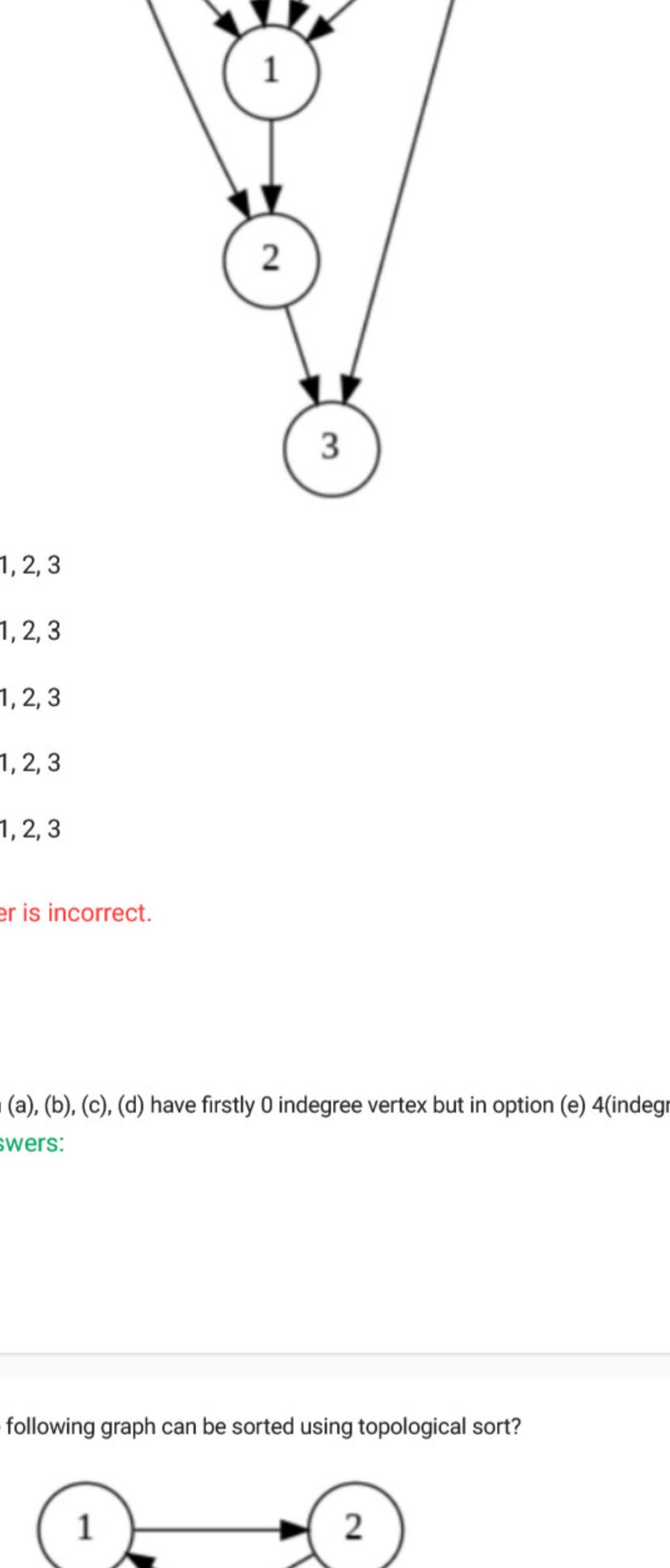
Feedback:

A complete graph has an edge between any two vertices. You can get an edge by picking any two vertices. So if there are n vertices, then $\binom{n}{2}$ which is option (d)

Accepted Answers:

$\frac{n(n-1)}{2}$

3) In the given directed graph, removing one edge e makes it a directed acyclic graph. Which of the following can be the possible values of e ? 1 point



- 1->3
- 2->4
- 7->4
- 4->1
- 1->2
- None, this is already direct acyclic graph.

No, the answer is incorrect.

Score: 0

Feedback:

Only one cycle is "1-2-4-1" any edges we can remove to make it Directed acyclic graph Hence option (b), (d), (e) are correct option.

Accepted Answers:

2->4
4->1
1->2

4) Given below is a function for traversing the graph using BFS(breadth-first search). For which of the following graphs will the function `BFS_adjList(vertices, AList)` always traverse the complete graph? 1 point

```
1 # 'vertices' is a list of vertices of graph 'G', and 'AList' is the
2 # adjacency list representation of graph 'G'.
3 # myqueue is a data structure, that has two methods 'enqueue(x)' and
4 # 'dequeue()' to add and remove(respectively)
5 # elements from queue in 'First in First out' manner.
6 # 'isEmpty()' method return True if Queue is empty, False otherwise.
7
8 def BFS_adjList(vertices, AList):
9     BFSList = []
10    q = myqueue()
11    visited = {v:0 for v in vertices}
12    q.enqueue(vertices[0])
13
14    while not q.isEmpty():
15        v = q.dequeue()
16        if not visited[v]:
17            BFSList.append(v)
18            for i in AList[v]:
19                if not visited[i]:
20                    q.enqueue(i)
21
22    return BFSList
```

- Connected directed graphs.
- Connected undirected graphs.
- Disconnected directed graphs.
- Disconnected undirected graphs.

No, the answer is incorrect.

Score: 0

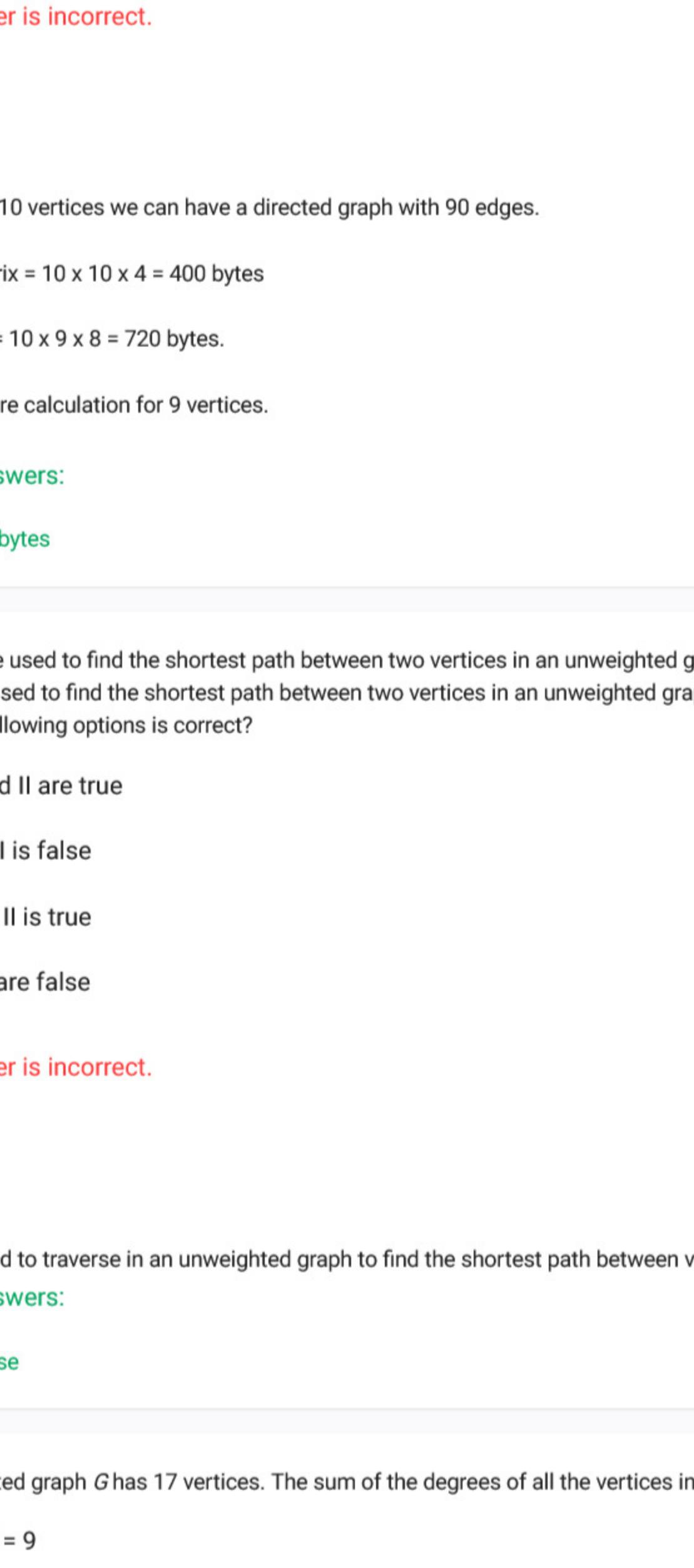
Feedback:

For disconnected graphs, this implementation will stop after traversing the connected component that contains the first vertex in the list of vertices. For connected directed graph this code will not work if our selected starting vertex does not have paths to every other vertex in the graph. Hence option (b) is correct.

Accepted Answers:

Connected undirected graphs.

5) Select all the possible topological sorted sequence(s) for the graph given below. 1 point



No, the answer is incorrect.

Score: 0

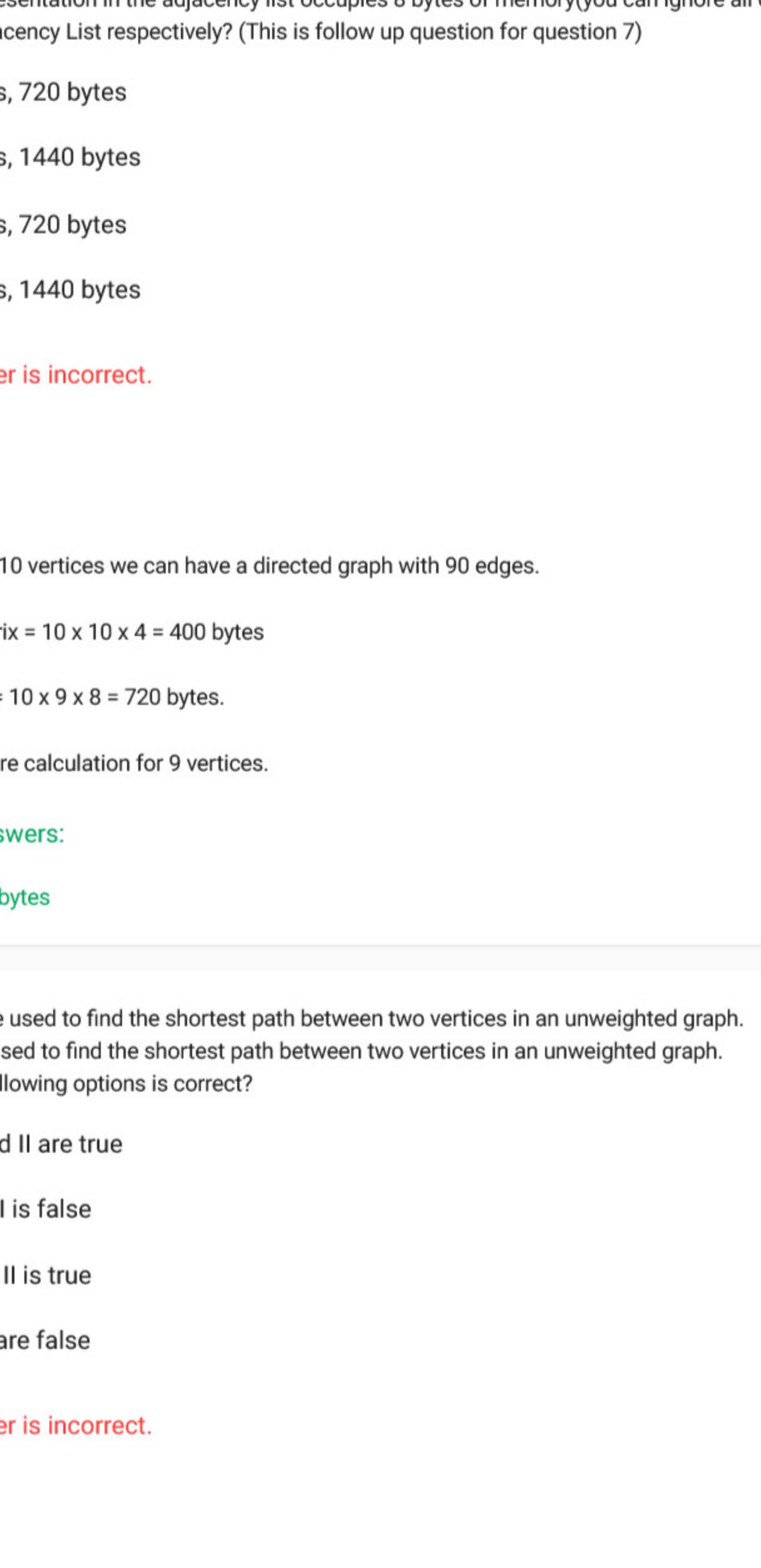
Feedback:

Topological sort works only for Directed Acyclic graph. Option (a), (b), (c), (d) have firstly 0 indegree vertex but in option (e) 4(indegree 1) came before 5(indegree 0) that is why it is incorrect.

Accepted Answers:

0, 5, 4, 6, 1, 2, 3
5, 6, 4, 0, 1, 2, 3
6, 0, 5, 4, 1, 2, 3
6, 5, 4, 0, 1, 2, 3

6) Which of the following graph can be sorted using topological sort? 1 point



No, the answer is incorrect.

Score: 0

Feedback:

Only BFS is used to traverse in an unweighted graph to find the shortest path between vertices in terms of the number of edges.

Accepted Answers:

0, 5, 4, 6, 1, 2, 3
5, 6, 4, 0, 1, 2, 3
6, 0, 5, 4, 1, 2, 3
6, 5, 4, 0, 1, 2, 3

7) Consider a directed graph G with 90 edges with the least number of vertices possible. What will be the number of vertices in graph G . 1 point

- 8
- 9
- 10
- 11

No, the answer is incorrect.

Score: 0

Feedback:

With minimum 10 vertices we can have a directed graph with 90 edges.

Accepted Answers:

0, 5, 4, 6, 1, 2, 3
5, 6, 4, 0, 1, 2, 3
6, 0, 5, 4, 1, 2, 3
6, 5, 4, 0, 1, 2, 3

8) We want to represent the graph G mentioned in previous question, in memory either as an Adjacency matrix or Adjacency list. Assume that each cell in the adjacency matrix takes 4 bytes of memory and each edge representation in the adjacency list occupies 8 bytes of memory(you can ignore all other factors that occupy memory). What will be the amount of memory required to represent graph G using Adjacency matrix and Adjacency List respectively? (This is follow up question for question 7) 1 point

- 324 bytes, 720 bytes
- 324 bytes, 1440 bytes
- 400 bytes, 720 bytes
- 400 bytes, 1440 bytes

No, the answer is incorrect.

Score: 0

Feedback:

All these option (a), (b), (c), (d) have firstly 0 indegree vertex but in option (e) 4(indegree 1) came before 5(indegree 0) that is why it is incorrect.

Accepted Answers:

0, 5, 4, 6, 1, 2, 3
5, 6, 4, 0, 1, 2, 3
6, 0, 5, 4, 1, 2, 3
6, 5, 4, 0, 1, 2, 3

9) BFS can be used to find the shortest path between two vertices in an unweighted graph. 1 point

If: DFS can be used to find the shortest path between two vertices in an unweighted graph.

Which of the following options is correct?

- Both I and II are true
- I is true, II is false
- I is false, II is true
- Both I, II are false

No, the answer is incorrect.

Score: 0

Feedback:

Only BFS is used to traverse in an unweighted graph to find the shortest path between vertices in terms of the number of edges.

Accepted Answers:

I is true, II is false

10) An undirected graph G has 17 vertices. The sum of the degrees of all the vertices in G is D . The number of vertices of even degree in G is K . Which of these values are possible for D and K ? 1 point

- $D = 42, K = 9$
- $D = 41, K = 9$
- $D = 42, K = 10$
- $D = 41, K = 10$

No, the answer is incorrect.

Score: 0

Feedback:

Sum of degree of all vertices in undirected graph = $2 \times$ number of edges hence it will be even number so $D=42$ if $D=42$ then number of edges will be 21. We know that undirected graph has even number of odd degree vertices.

Accepted Answers:

$D = 42, K = 9$

The due date for submitting this assignment has passed.

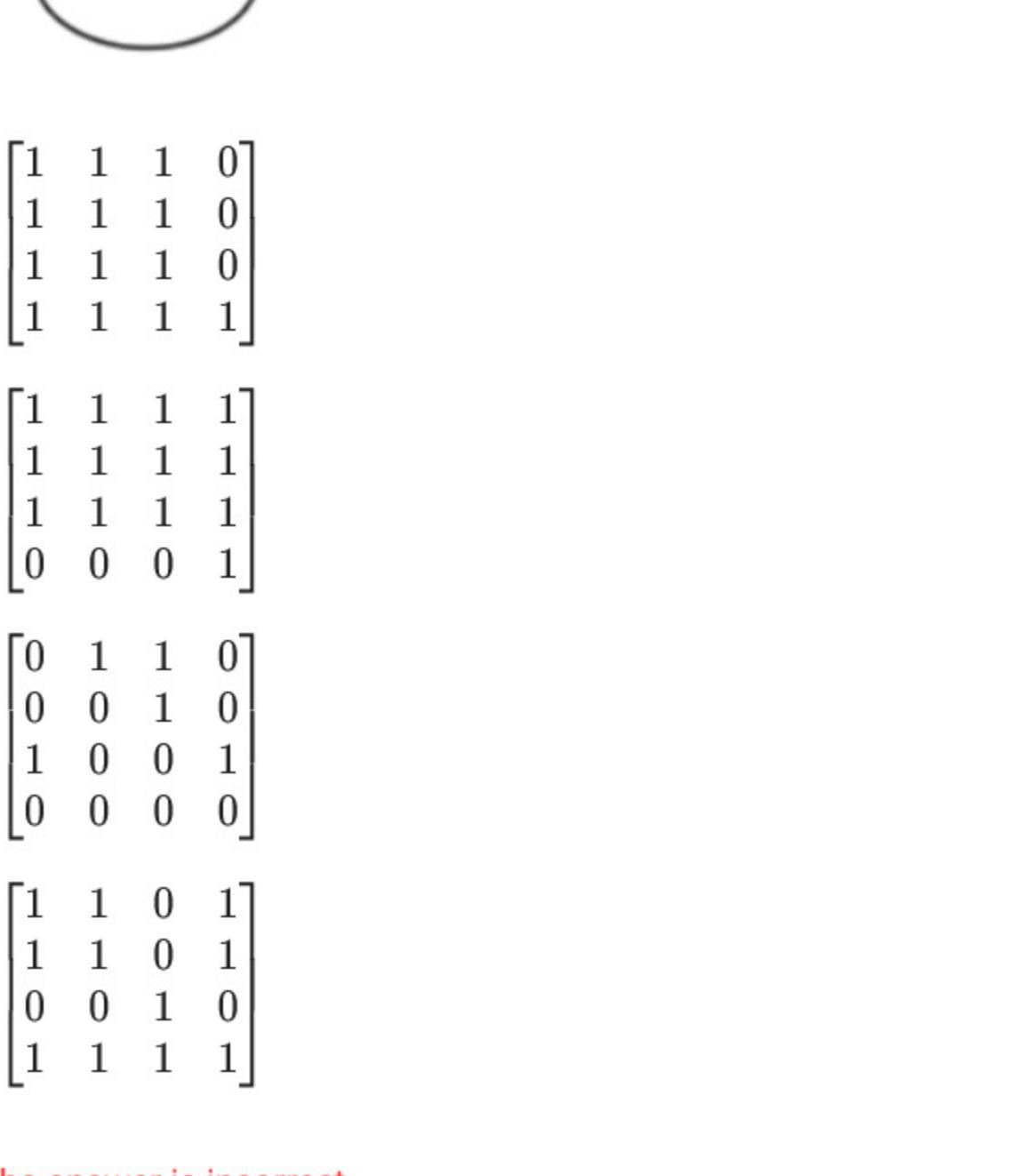
Due on 2022-06-12, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

This assignment will be evaluated after the deadline passes. You will get your score 48 hrs after the deadline. Until then the score will be shown as Zero.

1) Transitive closure(v)- If there is at least one path between vertex i to j or i == j then $v[i][j] = 1$ otherwise 0. What is the transitive closure for the given graph?

1 point



$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$

No, the answer is incorrect.

Score: 0

Feedback:

Transitive Closure

For all '(i,j)' pair of vertices in a graph, transitive closure matrix is formed by the reachability factor, i.e if 'j' is reachable from 'i' (means there is a path from 'i' to 'j') then we can put the matrix element as 1 or else if there is no path, then we can put it as 0. The reachability matrix is called the transitive closure of a graph. Second option is the correct transitive closure for the given graph

Accepted Answers:

$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

2) How can we use the Floyd-Warshall algorithm for all-pairs shortest paths to detect whether a graph has a negative cycle?

1 point

- Check if any shortest path entry $A[i][i]$ is negative.
 Check if any shortest path entry $A[i][i]$ is negative.
 Check if any shortest path entry $A[i][i]$ reduces from one iteration to the next.
 The Floyd-Warshall algorithm cannot be used to detect negative cycles.

No, the answer is incorrect.

Score: 0

Feedback:

If starting vertex and last vertex are same in a path called cycle. After completion of Floyd-Warshall algorithm, if ' $A[i][i]$ ' is negative, then it represents the graph has negative weight cycle. Hence, Second option is correct.

Accepted Answers:

Check if any shortest path entry $A[i][i]$ is negative.

3) Suppose we have a weighted undirected graph with a negative weight cycle. Which of the following is correct?

1 point

- Both Kruskal's algorithm and Prim's algorithm can be used to compute the minimum-cost spanning tree.
 Kruskal's algorithm will compute the minimum-cost spanning tree, but Prim's algorithm will not.
 Prim's algorithm will compute the minimum-cost spanning tree but, Kruskal's algorithm will not.
 Neither Kruskal's algorithm nor Prim's algorithm can be used to compute the minimum-cost spanning tree.

No, the answer is incorrect.

Score: 0

Feedback:

Only shortest path affected by negative weight cycle in graph. The concept of MST allows weights of an arbitrary sign. Both algorithms for finding MST (Kruskal's and Prim's) work fine with negative weight cycle or negative edges. Hence, first option is correct.

Accepted Answers:

Both Kruskal's algorithm and Prim's algorithm can be used to compute the minimum-cost spanning tree.

4) We can use Breadth First Search (BFS) instead of Dijkstra's algorithm to find out the shortest path from the given source node to every other node only _____.
1 point

- When all the edge weights are strictly greater than 1.
 When all the edge weights are distinct.
 When all the edge weights are equal.
 When graph has cycles.

No, the answer is incorrect.

Score: 0

Feedback:

BFS can be applied to find the shortest path in a graph in terms of number of edge in path from source vertex to destination vertex. So if all edges have equal weight, then we can apply BFS to find the shortest path (Number of edge in path * value of weight on single edge). Hence, third option is correct.

Accepted Answers:

When all the edge weights are equal.

5) Which of the following statement is/are true?

1 point

- Dijkstra's algorithm doesn't work for graphs with negative weights.
 Prim's algorithm doesn't work for graphs with negative weights.
 Floyd Warshall algorithm can detect negative weight cycle.
 Floyd Warshall algorithm works with negative weights but without negative cycle.

No, the answer is incorrect.

Score: 0

Feedback:

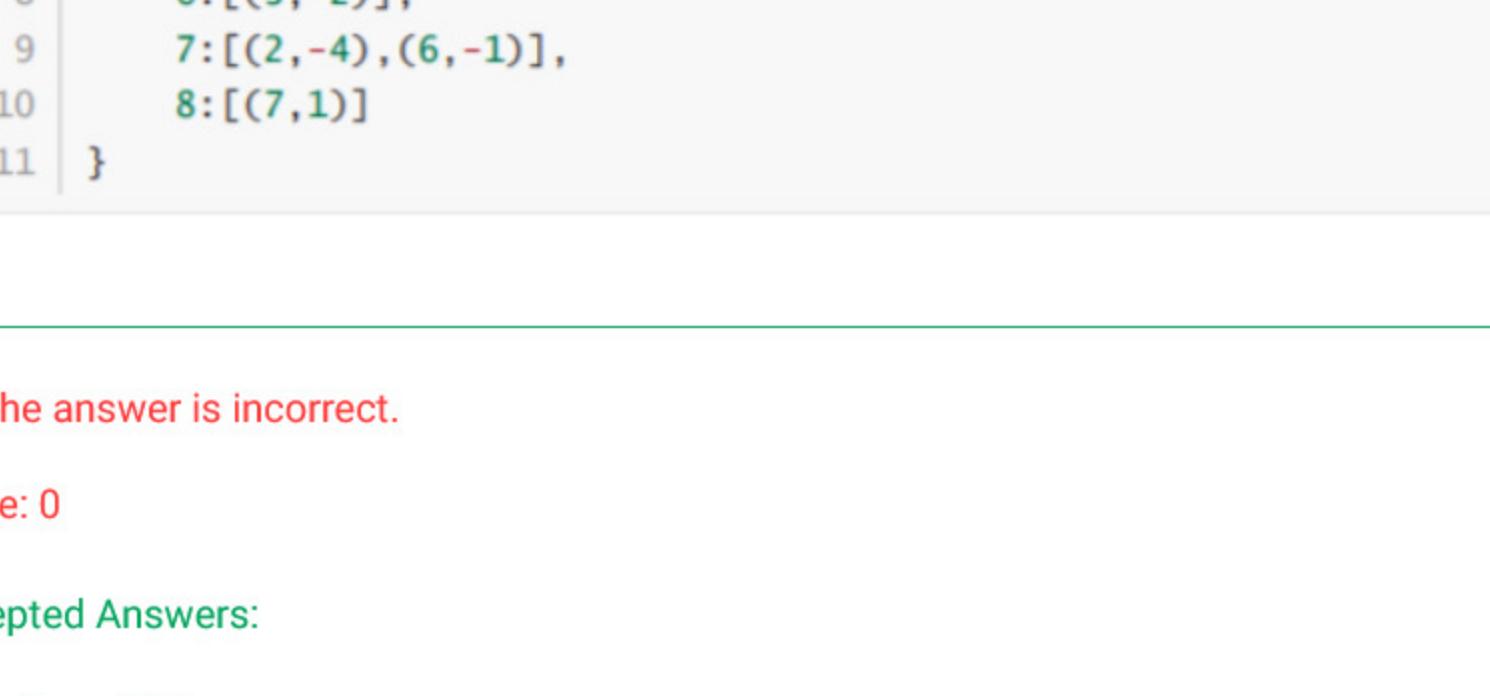
If some pair of edges in G have the same weight, then selection of edge can be different in Prim's and Kruskal's algorithm because of that minimum-cost spanning tree can be different with the same cost. If all edges in G have distinct weights, then always the same edges will be selected by both algorithm for MST. Hence, third option is correct

Accepted Answers:

Both (I) and (II) are correct.

7) Consider the graph shown below.

1 point



Which one of the following can be the sequence of edges added, in that order, to create a minimum spanning tree using Kruskal's algorithm?

- (a,b) (d,f) (b,f) (d,c) (d,e)
 (a,b) (d,f) (d,c) (b,f) (d,e)
 (d,f) (a,b) (d,c) (b,f) (d,e)
 (d,f) (a,b) (b,f) (d,c) (d,e)
 (d,f) (a,b) (b,f) (d,c) (b,c)

No, the answer is incorrect.

Score: 0

Feedback:

Given graph have many pair of edge with same weights, because of that more than one MST are possible based on the selection of edges. All options are representing correct sequence of edge of MST for given graph.

Accepted Answers:

(a,b) (d,f) (b,f) (d,c) (d,e)

(a,b) (d,f) (d,c) (b,f) (d,e)

(d,f) (a,b) (d,c) (b,f) (d,e)

(d,f) (a,b) (b,f) (d,c) (d,e)

(d,f) (a,b) (b,f) (d,c) (b,c)

Dijkstra's algorithm doesn't work for graphs with negative weights.

Floyd Warshall algorithm can detect negative weight cycle.

Floyd Warshall algorithm works with negative weights but without negative cycle.

6) Suppose we run Prim's algorithm and Kruskal's algorithm on a graph G and these two algorithms produce minimum-cost spanning trees T_P and T_K , respectively.

1 point

- (T_P) may be different from T_K if some pair of edges in G have the same weight.
 (T_P) is always the same as T_K if all edges in G have distinct weights.

Which of the following is true?

- Only (I) is correct.
 Only (II) is correct.
 Both (I) and (II) are correct.
 Both (I) and (II) are incorrect.

No, the answer is incorrect.

Score: 0

Feedback:

Minimum spanning tree does not affect by negative weights because of that second option is not true other than that all option are correct.

Accepted Answers:

Both (I) and (II) are correct.

8) Consider the given weighted adjacency matrix w for a complete undirected graph with vertex set $\{0, 1, 2, 3, 4\}$. Where $w[i][j]$, $i \neq j$ in the matrix is the weight of the edge (i,j) .

1 point

$$w = \begin{pmatrix} 0 & 1 & 8 & 1 & 4 \\ 1 & 0 & 12 & 4 & 9 \\ 8 & 12 & 0 & 7 & 3 \\ 1 & 4 & 7 & 0 & 2 \\ 4 & 9 & 3 & 2 & 0 \end{pmatrix}$$

What is the weight of the minimum spanning tree for the given graph?

1 point

No, the answer is incorrect.

Score: 0

Feedback:

Given graph have many pair of edge with same weights, because of that more than one MST are possible based on the selection of edges. All options are representing correct sequence of edge of MST for given graph.

Accepted Answers:

(a,b) (d,f) (b,f) (d,c) (d,e)

(a,b) (d,f) (d,c) (b,f) (d,e)

(d,f) (a,b) (d,c) (b,f) (d,e)

(d,f) (a,b) (b,f) (d,c) (d,e)

(d,f) (a,b) (b,f) (d,c) (b,c)

Dijkstra's algorithm doesn't work for graphs with negative weights.

Floyd Warshall algorithm can detect negative weight cycle.

Floyd Warshall algorithm works with negative weights but without negative cycle.

10) Consider the following weighted adjacency list $WList$ for a directed and connected graph. What will be the path weight of the shortest path from 1 to 3?

```
1 wList = {  
2     #source:[destination,weight],...  
3     1:[(2,10),(8,8)],  
4     2:[(6,2)],  
5     3:[(2,1),(4,1)],  
6     4:[(5,3)],  
7     5:[(6,-1)],  
8     6:[(3,-2)],  
9     7:[(2,-4),(6,-1)],  
10    8:[(7,1)]  
11 }
```

1 point

No, the answer is incorrect.

Score: 0

Feedback:

All options are representing correct properties of MST.

Accepted Answers:

A spanning tree is a connected acyclic graph.

A spanning tree for an n vertex graph has exactly n-1 edges.

Adding an edge to a spanning tree must create a cycle.

In a spanning tree, every pair of nodes is connected by a unique path.

No, the answer is incorrect.

Score: 0

Feedback:

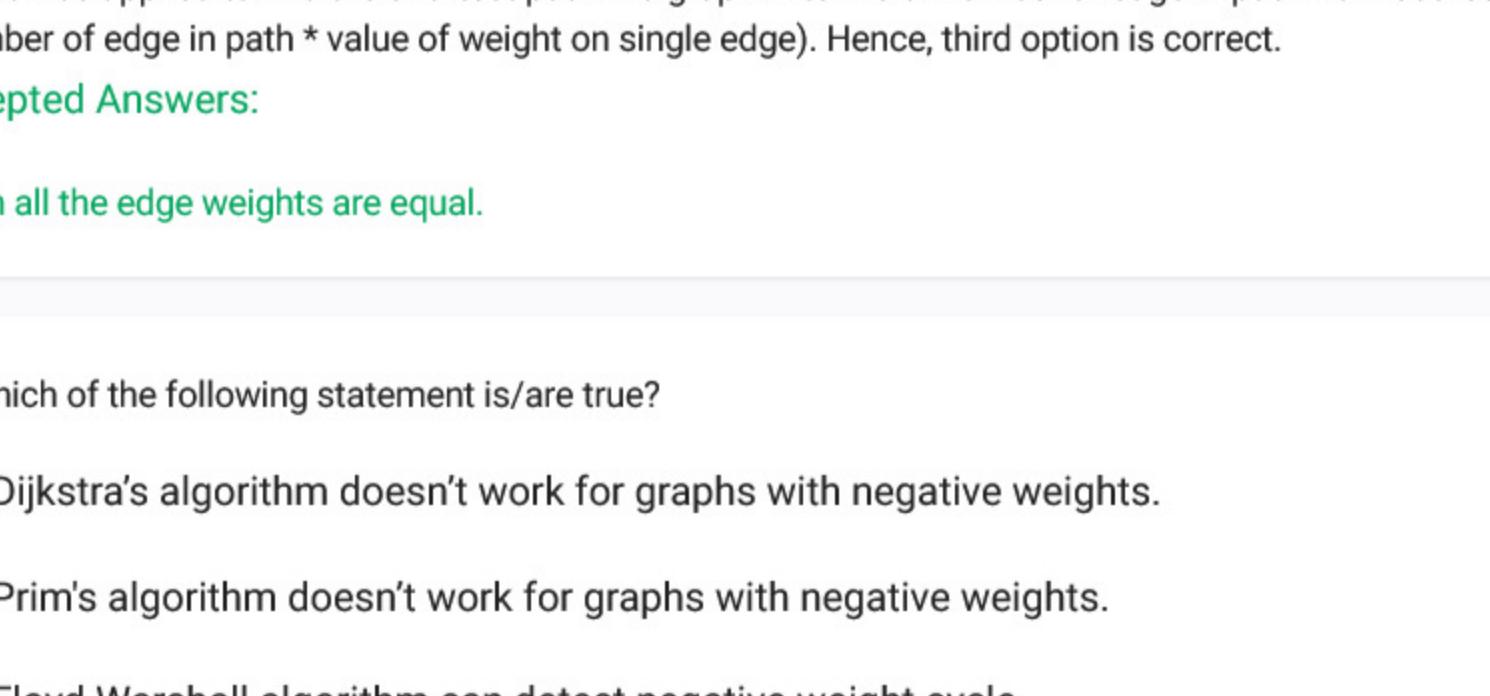
If some pair of edges in G have the same weight, then selection of edge can be different in Prim's and Kruskal's algorithm because of that minimum-cost spanning tree can be different with the same cost. If all edges in G have distinct weights, then always the same edges will be selected by both algorithm for MST. Hence, third option is correct

Accepted Answers:

Both (I) and (II) are correct.

7) Consider the graph shown below.

1 point



Which one of the following can be the sequence of edges added, in that order, to create a minimum spanning tree using Kruskal's algorithm?

- (a,b) (d,f) (b,f) (d,c) (d,e)
 (a,b) (d,f) (d,c) (b,f) (d,e)
 (d,f) (a,b) (d,c) (b,f) (d,e)
 (d,f) (a,b) (b,f) (d,c) (d,e)
 (d,f) (a,b) (b,f) (d,c) (b,c)

No, the answer is incorrect.

Score: 0

Feedback:

Given graph have many pair of edge with same weights, because of that more than one MST are possible based on the selection of edges. All options are representing correct sequence of edge of MST for given graph.

Accepted Answers:

(a,b) (d,f) (b,f) (d,c) (d,e)

(a,b) (d,f) (d,c) (b,f) (d,e)

(d,f) (a,b) (d,c) (b,f) (d,e)

(d,f) (a,b) (b,f) (d,c) (d,e)

(d,f) (a,b) (b,f) (d,c) (b,c)

The due date for submitting this assignment has passed.

Due on 2022-06-15, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

Note: This assignment will be evaluated after the deadline passes. You will get your score 48 hrs after the deadline. Until then the score will be shown as Zero.

1) Consider a list L with k distinct numbers and a heap H with size n . What is the nearest upper bound for checking if every number in L is present in the heap H ? 1 point

- $O(kN)$
- $O(k \log N)$
- $O(N \log k)$
- $O(\log N \log k)$

No, the answer is incorrect.

Score: 0

Feedback:

For searching any element in heap you need to traverse all the elements of the heap or in some cases $O(N/2)$ but ultimately we come to $O(N)$ and for 'k' distinct element it will be $O(kN)$.

Accepted Answers:

$O(kN)$

Use the below information for the Questions 2 to 7

Identify the below arrays as min heap, max heap or none. Type *min* if it is a min heap, *max* if it is a max heap, *none* otherwise.

2)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|

1 point

- min
- max
- none

No, the answer is incorrect.

Score: 0

Feedback:

When we choose the element from the array in the same sequence and construct the heap, it will be min-heap.

Accepted Answers:

min

3)

| | | | | | | | | |
|----|----|----|----|---|---|---|----|---|
| 67 | 65 | 43 | 54 | 6 | 2 | 1 | 19 | 5 |
|----|----|----|----|---|---|---|----|---|

1 point

min
 max
 none

No, the answer is incorrect.

Score: 0

Feedback:

When we choose the element from the array in the same sequence and construct the heap, it will be max-heap.

Accepted Answers:

max

4)

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 88 | 77 | 66 | 55 | 44 | 33 | 22 | 11 |
|----|----|----|----|----|----|----|----|

1 point

- min
- max
- none

No, the answer is incorrect.

Score: 0

Feedback:

When we choose the element from the array in the same sequence and construct the heap, it will be max-heap.

Accepted Answers:

max

5)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 8 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|

1 point

- min
- max
- none

No, the answer is incorrect.

Score: 0

Feedback:

When we choose the first element of the array is neither minimum nor maximum hence it is neither min-heap nor max-heap.

Accepted Answers:

none

7) Which of the following will correctly represent the max-heap, after inserting elements 1, 2, 3, 5, 7, 6 and 4 in the given order, starting with an empty heap? 1 point

- | | | | | | | |
|---|---|---|---|---|---|---|
| 7 | 5 | 6 | 1 | 3 | 2 | 4 |
|---|---|---|---|---|---|---|
- | | | | | | | |
|---|---|---|---|---|---|---|
| 7 | 5 | 6 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
- | | | | | | | |
|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|

No, the answer is incorrect.

Score: 0

Feedback:

After constructing the max heap by the given elements of the sequence, then option (a) is correct.

Accepted Answers:

O(n), O(log n)

8) What is the nearest worst case upper bound to search for an element in a heap and in a binary search tree, respectively? Consider that the total number of elements are n and the binary search tree is balanced (in the form of a complete binary tree). 1 point

- $O(\log n), O(\log n)$
- $O(n), O(n)$
- $O(n), O(\log n)$
- $O(\log n), O(n)$

No, the answer is incorrect.

Score: 0

Feedback:

To remove the max element from the max-heap, we can swap the first element(max) to the last element then remove the last element in $O(1)$ time, after that we have to perform heapify on the first element it will take $O(\log n)$ time to create max heap again. So overall complexity will be $O(\log n)$ or $O(n)$.

Accepted Answers:

O(n), O(log n)

9)

```
1 def max_heapify(A,k):  
2     l = 2 * k + 1  
3     r = 2 * k + 2  
4     largest = k  
5     if l < len(A) and A[l] > A[k]:  
6         largest = l  
7     if r < len(A) and A[r] > A[largest]:  
8         largest = r  
9     if largest == k:  
10        A[k], A[largest] = A[largest], A[k]  
11        max_heapify(A, k)  
12  
13 def build_max_heap(A):  
14     n = int((len(A)/2)-1)  
15     for k in range(n, -1, -1):  
16         max_heapify(A, k)
```

1 point

Above code is an incorrect implementation of *max_heap*. Which of the following lines of code if replaced for the given line numbers will correct the above code?

- ```
1 if largest != k: #line 9
2 max_heapify(A,largest) #line 11
```
- ```
1 | max_heapify(A,largest) #line 11
```
- ```
1 max_heapify(A,largest) #line 11
2 for k in range(n): #line 15
```
- ```
1 max_heapify(A,largest) #line 11  
2 for k in range(n,-1, 0): #line 15
```

No, the answer is incorrect.

Score: 0

Feedback:

Here if the largest index is not equal to index k at which heapify takes place then we will swap the largest index element to kth element, and then in line 11, we will call heapify function at largest index because now here at largest index has an incorrect number, so we need to heapify it. Hence, option (a) is correct.

Accepted Answers:

O(n), O(log n)

10) Consider a max heap, represented as the array

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|---|----|
| 40 | 30 | 18 | 20 | 15 | 16 | 17 | 10 | 4 | 25 |
|----|----|----|----|----|----|----|----|---|----|

. Now consider that a value 25 is inserted into this heap. After insertion, the new heap will be 1 point

- | | | | | | | | | | |
|----|----|----|----|----|----|----|----|---|----|
| 40 | 30 | 18 | 20 | 15 | 16 | 17 | 10 | 4 | 15 |
|----|----|----|----|----|----|----|----|---|----|
- | | | | | | | | | | |
|----|----|----|----|----|----|----|----|---|----|
| 40 | 30 | 18 | 20 | 16 | 25 | 17 | 10 | 4 | 15 |
|----|----|----|----|----|----|----|----|---|----|
- | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|---|
| 40 | 30 | 18 | 20 | 15 | 25 | 16 | 17 | 10 | 4 |
|----|----|----|----|----|----|----|----|----|---|

No, the answer is incorrect.

Score: 0

Feedback:

Insertion of an element in heap takes place at the lower level and leftmost place and then by heapify, we restore max heap property step by step from leaf node to the root node and only option (b) is satisfying this property.

Accepted Answers:

40 | 30 | 18 | 20 | 25 | 16 | 17 | 10 | 4 | 15

11) Pre-Order traversal of binary search tree is 15,10,12,11,20,18,16,19. Which one of the following is the post-order traversal of the binary search tree? 1 point

- 10,11,12,15,18,19,20,18,15
- 11,12,10,16,19,20,18,20,15
- 11,12,10,16,19,18,20,15
- 11,12,10,20,15,18,19,16

No, the answer is incorrect.

Score: 0

Feedback:

"traverse" function is traversing the left child first then the right child and then traversing the root node hence it is post-order traversal.

Accepted Answers:

11,12,10,16,19,20,18,15

12) What form of tree traversal does the function *traverse(root)* implement, where *root* is the root node of a binary search tree? 1 point

```
1 class Node:  
2     def __init__(self, data):  
3         self.data = data  
4         self.left = None  
5         self.right = None  
6  
7     def traverse(self):  
8         if self == None:  
9             return  
10        traverse(self.left)  
11        traverse(self.right)  
12        print(self.data, end=" ")
```

Above code is an incorrect implementation of *traverse*. Which of the following lines of code if replaced for the given line numbers will correct the above code?

- ```
1 if largest != k: #line 9
2 max_heapify(A,largest) #line 11
```
- ```
1 | max_heapify(A,largest) #line 11
```
- ```
1 max_heapify(A,largest) #line 11
2 for k in range(n): #line 15
```
- ```
1 max_heapify(A,largest) #line 11  
2 for k in range(n,-1, 0): #line 15
```

No, the answer is incorrect.

Score: 0

Feedback:

After constructing the max heap by the given elements of the sequence, then option (a) is correct.

Accepted Answers:

*1 if largest != k: #line 9
2 max_heapify(A,largest) #line 11*

13) Consider a max-heap H with n elements and h height. What is the nearest upper bound to remove the maximum element from max-heap H ? 1 point

- $O(h)$
- $O(n)$
- $O(n), O(\log n)$
- $O(\log n), O(n)$

No, the answer is incorrect.

Score: 0

Feedback:

To remove the max element from the max-heap, we can swap the first element(max) to the last element then remove the last element in $O(1)$ time, after that we have to perform heapify on the first element it will take $O(\log n)$ time to create max heap again. So overall complexity will be $O(\log n)$ or $O(n)$.

Accepted Answers:

O(n), O(log n)

14) The given code implements a priority queue using a sorted list. Which of the following is the correct implementation of *delete_max* operation of an element from the priority queue? Note that the higher the priority the higher its priority? 1 point

```
1 class PriorityQueue:  
2     def __init__(self):  
3         self.que = []
```

Above code is an incorrect implementation of *delete_max*. Which of the following lines of code if replaced for the given line numbers will correct the above code?

- ```
1 if largest != 0: #line 9
2 del self.que[largest] #line 11
```
- ```
1 | del self.que[largest] #line 11
```
- ```
1 max_heapify(A,largest) #line 11
2 for k in range(n): #line 15
```
- ```
1 max_heapify(A,largest) #line 11  
2 for k in range(n,-1, 0): #line 15
```

No, the answer is incorrect.

Score: 0

Feedback:

Here the larger the element the larger its priority, so we are trying to find out the index of the largest element in the queue and then storing the index of it in the largest variable and then deleting the largest element, and option (c) is performing only this activity.

Accepted Answers:

*1 if largest != 0: #line 9
2 del self.que[largest] #line 11*

15) Consider a max-heap H with n elements and h height. What is the nearest upper bound to remove the maximum element from max-heap H ? 1 point

The due date for submitting this assignment has passed.

Due on 2022-06-22, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

Note: This assignment will be evaluated after the deadline passes. You will get your score 48 hrs after the deadline. Until then the score will be shown as Zero.

1)



1 point

Which of the following insertion order of data items will generate the above AVL tree?

- 1,2,3,4,5,6,7
- 3,2,6,1,5,4
- 7,6,5,4,3,2,1
- 5,3,6,2,4,7,1

No, the answer is incorrect.

Score: 0

Feedback:

Insert nodes one by one in a given sequence according to the binary search rule and update the slope or balance factor (absolute difference between left height and right height) of each node. If at any time the slope value of any node is more than 1 then apply left or right rotation to balance the tree. Create the AVL tree for given sequences, after that, you can observe options (a) and (c) will generate the given AVL tree.

Accepted Answers:

1,2,3,4,5,6,7
7,6,5,4,3,2,1

2)

```
1 class AVLTree:  
2     # Constructor:  
3     def __init__(self,initial = None):  
4         self.value = initial  
5         if self.value:  
6             self.left = AVLTree()  
7             self.right = AVLTree()  
8             self.height = 1  
9         else:  
10            self.left = None  
11            self.right = None  
12            self.height = 0
```

1 point

For the above class `AVLTree`, if the leaf node has height 1, then which of the following options will compute the height of all nodes in the tree? Consider that `self` is referencing the root node of the tree.

- `def computeheight(self):
 if self.isEmpty():
 return
 else:
 self.height = 1 + max(self.left.height, self.right.height)
 self.left.computeheight()
 self.right.computeheight()`
- `def computeheight(self):
 if self.isEmpty():
 return
 else:
 self.left.computeheight()
 self.height = 1 + max(self.left.height, self.right.height)
 self.right.computeheight()`

- `def computeheight(self):
 if self.isEmpty():
 return
 else:
 self.height = 1 + max(self.left.height, self.right.height)
 self.left.computeheight()
 self.right.computeheight()`
- `def computeheight(self):
 if self.isEmpty():
 return
 else:
 self.height = 1 + max(self.left.height, self.right.height)
 self.left.computeheight()
 self.right.computeheight()`

No, the answer is incorrect.

Score: 0

Feedback:

We can use post-order traversal to compute the height of each node because, in post-order traversal, we traverse from a leaf node, after visiting the left child and the right child we traverse the parent node. So during traversing the parent node, we can calculate the height using '`self.height = 1 + max(self.left.height, self.right.height)`'. Hence, option(c) is the correct implementation to calculate the height of each node.

Accepted Answers:

```
1 def computeheight(self):  
2     if self.isEmpty():  
3         return  
4     else:  
5         self.left.computeheight()  
6         self.right.computeheight()  
7         self.height = 1 + max(self.left.height, self.right.height)
```

3) A manager claims scheduling jobs based on the slack time to time taken ratio $\left(\frac{D(i)-T(i)}{T(i)}\right)$ is optimum. His strategy is to complete the job which has the least slack to the time taken ratio. Choose the counter-example to prove the strategy is not optimum.

- T[1] = 2; D[1] = 15;
T[2] = 4; D[2] = 9
- T[1] = 9; D[1] = 10;
T[2] = 2; D[2] = 2
- T[1] = 1; D[1] = 10;
T[2] = 2; D[2] = 3
- T[1] = 9; D[1] = 15;
T[2] = 4; D[2] = 9

No, the answer is incorrect.

Score: 0

Feedback:

In option (d), the slack time to time taken ratio for job 1 is less than compared to job 2, if we follow this order (job 1 then job 2) then the lateness of job 2 will be 4, but if we follow the reverse order (job 2 then job 1) then the lateness of both jobs will be 0. So we can say that this type of scheduling strategy is not optimum.

Accepted Answers:

T[1] = 9; D[1] = 15;
T[2] = 4; D[2] = 9

4) In the Activity Selection Problem, each activity i has a start time S_i and a finish time F_i where $S_i < F_i$. Two or more activities can not perform simultaneously. Activity i and j are said to be non-conflicting if _____.

- $S_i \leq F_j$
- $S_j \leq F_i$
- $S_i \geq F_j$ or $S_j \geq F_i$
- $S_i \leq S_j$ and $F_i \leq F_j$

No, the answer is incorrect.

Score: 0

Feedback:

Frequency(`probability`) is given for each character. We can generate Huffman codes for each character. After that, we can calculate the length of code for each character and multiply this by the same probability ratio of characters in a 1000 characters message.

$A = 2(\text{length of Huffman code}) * 220(\text{frequency in message(1000 character)}) \text{ in the ratio of probability}) = 440$

B = 2 * 340 = 680

C = 3 * 170 = 510

D = 2 * 190 = 380

E = 3 * 80 = 240

Total = 2250

Accepted Answers:

2250

5) In the table below, we have 8 activities with the corresponding start and finish times. It might not be possible to complete all the activities since their time frame can conflict. For example, if any activity starts at time 0 and finishes at time 4, then other activities can not start before 4. It can be started at 4 or afterwards. What is the maximum number of activities which can be performed without conflict?

| Activity | Start time | Finish time |
|----------|------------|-------------|
| A | 1 | 3 |
| B | 3 | 4 |
| C | 0 | 7 |
| D | 1 | 2 |
| E | 5 | 6 |
| F | 5 | 9 |
| G | 10 | 11 |
| H | 7 | 8 |

No, the answer is incorrect.

Score: 0

Accepted Answers:

(Type: Numeric) 5

1 point

6) A machine has to rest for an hour mandatorily after every 5 hours of continuous work but the machine can take rest for an hour before completion of 5 hours and become available for next 5 hours. What will be the value of lateness of the given list of jobs in an optimal sequence?

$$\text{Lateness} = \sum_{i=0}^3 L(i)$$

$$L(i) = \begin{cases} \text{if } \text{TimeDelivered}(i) > \text{TimeDue}(i), \text{ TimeDelivered}(i) - \text{TimeDue}(i) \\ \text{else}, 0 \end{cases}$$

| Id | Time Required | Due |
|----|---------------|-----|
| 0 | 2 | 10 |
| 1 | 1 | 2 |
| 2 | 4 | 8 |
| 3 | 3 | 7 |

1 point

No, the answer is incorrect.

Score: 0

Accepted Answers:

10101101101000010011

11011000110011001001

101010110101000010001

11001110001000101000

11001110001011100011

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

11010111001011100011001

The due date for submitting this assignment has passed.

Due on 2022-07-03, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

- 1) In an array A , two elements $A[i]$ and $A[j]$ form an inversion pair, if $A[i] > A[j]$ for $i < j$.

1 point

Which of the following input arrays will have maximum and minimum inversion pairs respectively?

- I. Array sorted in ascending order.
II. Array sorted in descending order.

- I and II
 II and I
 I and I
 II and II

No, the answer is incorrect.

Score: 0

Feedback:

Suppose an array contains 'n' elements. If the array is sorted in ascending order then the number of inversion pairs will be 0 and if the array is sorted in descending order then the number of inversion pair will be ' $n(n-1)/2$ '. Hence, option (b) is the correct answer.

Accepted Answers:

II and I

Question 2 & 3

Consider the following strategy to solve a problem of input size n .

Divide the problem into 6 sub-problems, each of size $\frac{n}{6}$. Number of steps required to combine these 6 solutions is $2n + 12$. We apply this strategy recursively till the sub-problems can not be further divided into sub-problems.

- 2) What will be the nearest upper bound for the above algorithm?

1 point

- $O(n^2)$
 $O(n^6)$
 $O(n \log n)$
 $O(n)$

No, the answer is incorrect.

Score: 0

Feedback:

The recurrence relation will be $T(n) = 6T(n/6) + O(n)$, so the depth of the tree will be $\log_6 n$ and each level complexity is $O(n)$ to combine the solution, then overall complexity will be $O(n \log_6 n) = O(n \log n)$. Hence, option (c) is correct.

Accepted Answers:

$O(n \log n)$

- 3) If we divide the problem into 4 sub-problems of size $\frac{n}{2}$ and number of steps required to combine the solutions is 15 using some optimizations, what will be the nearest upper bound of this algorithm?

1 point

- $O(n^2)$
 $O(n^4)$
 $O(\log n)$
 $O(n)$

No, the answer is incorrect.

Score: 0

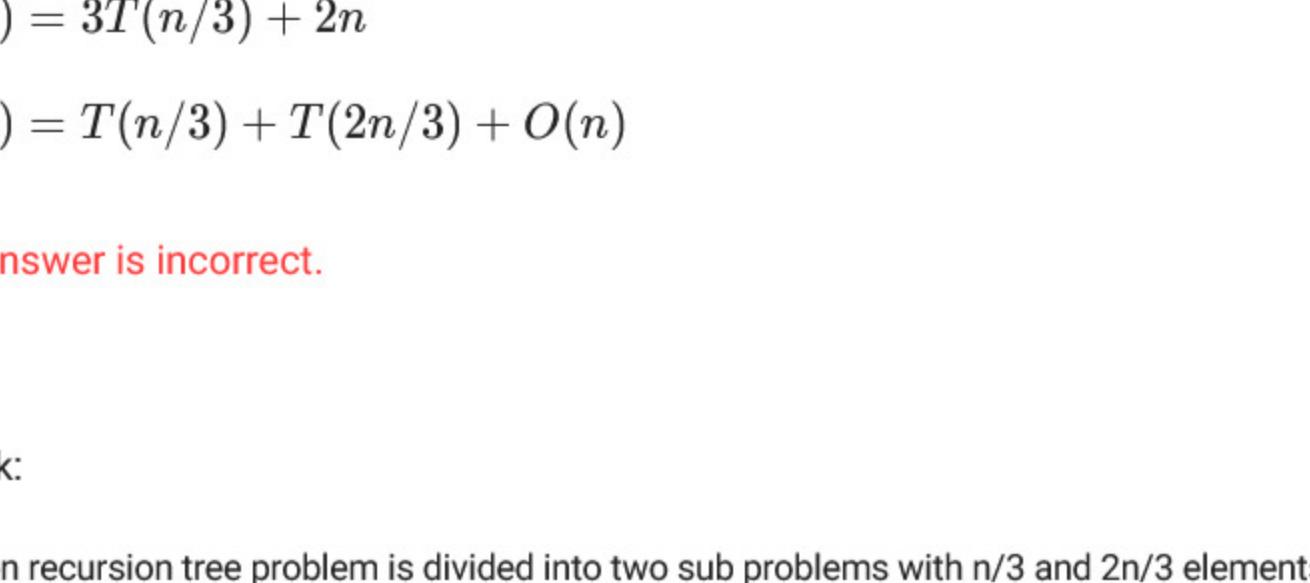
Feedback:

If series grows exponentially then recurrence relation will be $T(n) = 4T(n/2) + O(1)$ then we can use formula $T(n) = O(n^{\log_r c})$ for recursion tree. Here $r = 4$, $c=2$, $T(n) = O(n^{\log_2 4}) = O(n^2)$. Hence, option (a) is correct.

Accepted Answers:

$O(n^2)$

Question 4 & 5



- 4) Which of the following options describe the recurrence relation that will give the above recurrence tree, if the cost to combine solutions at each level is equal to $2n$ and the cost to compute the solution of a sub-problem at leaf nodes is $\Theta(1)$?

- $T(n) = T(n/3) + T(2n/3) + 2n$
 $T(n) = T(n/3) + T(2n/3) + n$
 $T(n) = 3T(n/3) + 2n$
 $T(n) = T(n/3) + T(2n/3) + O(n)$

No, the answer is incorrect.

Score: 0

Feedback:

In the given recursion tree problem is divided into two sub problems with $n/3$ and $2n/3$ elements and combine solution takes $2n$ cost at each level so recurrence relation will be $T(n) = T(n/3) + T(2n/3) + 2n$, or it can also represent $T(n) = T(n/3) + T(2n/3) + O(n)$. Hence, option (a) and (d) both are correct.

Accepted Answers:

$T(n) = T(n/3) + T(2n/3) + 2n$
 $T(n) = T(n/3) + T(2n/3) + O(n)$

- 5) If $O(f(n))$ is the upper bound for the above recurrence relation. What is the value of $f(n)$?

1 point

- $\log n$
 $n \log n$
 n^2
 $n^{\frac{3}{2}} \log n$

No, the answer is incorrect.

Score: 0

Feedback:

The recurrence relation is $T(n) = T(n/3) + T(2n/3) + O(n)$, so depth of tree will be $\log_{3/2} n$ and each level complexity is $O(n)$ to combine solution, then overall complexity will be $O(n \log_{3/2} n) = O(n \log n)$. Hence, option (c) is correct.

Accepted Answers:

$n \log n$

The due date for submitting this assignment has passed.

Due on 2022-07-03, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

1) Consider product of three matrices A, B and C having size $p \times q$, $q \times r$ and $r \times s$ respectively. Which of the following statements is/are true? 1 point

- Computing $(AB)C$ and $A(BC)$ both always take equal time.
- Computing $(AB)C$ and $A(BC)$ both take equal time if $p = q = r = s$.
- If $(1/q + 1/s) < (1/p + 1/r)$ then $(AB)C$ takes less time than $A(BC)$.
- $(AB)C$ takes $pr(q+s)$ steps to compute.
- $(AB)C$ takes less time than $A(BC)$ if $(p+q) > (r+s)$.

No, the answer is incorrect.

Score: 0

Accepted Answers:

Computing $(AB)C$ and $A(BC)$ both take equal time if $p = q = r = s$.
If $(1/q + 1/s) < (1/p + 1/r)$ then $(AB)C$ takes less time than $A(BC)$.
 $(AB)C$ takes $pr(q+s)$ steps to compute.

2) Consider four matrices M_1, M_2, M_3 , and M_4 of dimension $p \times q, q \times r, r \times s$, and $s \times t$ respectively, where p, q, r and s are distinct values. In How many ways can we evaluate the product of M_1, M_2, M_3 , and M_4 if each step consist of multiplying two matrices? 1 point

- 3
- 5
- 7
- 9

No, the answer is incorrect.

Score: 0

Accepted Answers:

5

3) Consider the following function `f(seq)`, where `seq` is a list of integers. 1 point

```
1 def f(seq):
2     n = len(seq)
3     L=[0]*(n-1)+[1]
4     for i in range(n-2,-1,-1):
5         if seq[i] < seq[i+1]:
6             L[i]= 1 + L[i+1]
7         else:
8             L[i] = 1
9     return(max(L))
```

Which of the following statements is/are true ?

- The algorithm uses the divide and conquer paradigm.
- The function returns the length of the longest contiguous increasing sequence of numbers in the list.
- The function takes $O(n)$ to return the output.
- The function returns the length of the longest increasing subsequence of the number in the list.
- The algorithm uses the dynamic programming paradigm.

No, the answer is incorrect.

Score: 0

Accepted Answers:

The function returns the length of the longest contiguous increasing sequence of numbers in the list.
The function takes $O(n)$ to return the output.
The algorithm uses the dynamic programming paradigm.

4) Which of the following statements is/are true about the edit distance problem? 1 point

- It can be solved using the dynamic programming method.
- It can be solved using dynamic programming in $O(m+n)$ time, where m and n are sizes of two strings.
- The edit distance will be zero only when the two strings are equal.
- The maximum edit distance between the two strings is equal to the length of the larger string.

No, the answer is incorrect.

Score: 0

Accepted Answers:

$O(m^n), O(mn)$

5) What is the length of the longest common subsequence of strings `ABCBDABACD` and `BDCABADCD`? 1 point

- 3
- 5
- 6
- 7

No, the answer is incorrect.

Score: 0

Accepted Answers:

$O(m^n), O(mn)$

Questions 8 to 10 are based on the common theme

The subset sum problem is defined as follows. Given a list L of n non-negative integers and a value k , determine if there is a subset of elements from the list L whose sum is equal to k . Consider the following solution code where T is a 2-dimensional Boolean array, with $n+1$ rows and $k+1$ columns. $T[i][j]$, $1 \leq i \leq n$, $1 \leq j \leq k$ is *True*, if and only if there is a subset of $\{a_1, a_2, \dots, a_n\}$ whose sum is equal to j .

```
1 def subsetSum(L, k):
2     n = len(L)
3     T = [[False for x in range(k + 1)] for y in range(n + 1)]
4     for i in range(n + 1):
5         T[i][0] = True
6     for i in range(1, n + 1):
7         for j in range(1, k + 1):
8             if L[i - 1] > j:
9                 T[i][j] = T[i - 1][j]
10            else:
11                T[i][j] = T[i - 1][j] or T[i - 1][j - L[i - 1]]
12    return _____
```

What will be the asymptotic complexity of the above recursive function without and with memoization respectively? 1 point

- $O(m^n), O(mn)$
- $O(2^n), O(n^2)$
- $O(n^m), O(n)$
- $O(2^{m+n}), O(m+n)$

No, the answer is incorrect.

Score: 0

Accepted Answers:

$O(m^n), O(mn)$

7) The number n is to be created by adding the elements(one element can be used more than one time) of a list of positive integers L of length m . 1 point

Feedback:

After filling the table T completely, the Boolean value at position $T[n][k]$ represent that there is a subset of elements whose sum is equal to k or not.

Accepted Answers:

$T[n][k]$

8) Function `subsetSum()` is an example of. 1 point

- A greedy algorithm
- A dynamic programming algorithm
- A divide and conquer algorithm
- None of these

No, the answer is incorrect.

Score: 0

Feedback:

We can calculate the complexity using nested for loop line numbers 6 and 7. So complexity will be $O(nk)$.

Accepted Answers:

$O(nk)$

9) Function `subsetSum()` is an example of. 1 point

- A greedy algorithm
- A dynamic programming algorithm
- A divide and conquer algorithm
- None of these

No, the answer is incorrect.

Score: 0

Feedback:

Here we are storing the solution of the sub problem in table T and using this solution to calculate the solution of further remaining problems. That means the algorithm follows the dynamic programming approach.

Accepted Answers:

$A \text{ dynamic programming algorithm}$

10) What is the time complexity of function `subsetSum()`? 1 point

- $O(n^2)$
- $O(n \log n)$
- $O(n+k)$
- $O(nk)$

No, the answer is incorrect.

Score: 0

Feedback:

We can calculate the complexity using nested for loop line numbers 6 and 7. So complexity will be $O(nk)$.

Accepted Answers:

$O(nk)$

The due date for submitting this assignment has passed.

Due on 2022-07-17, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

1 point

1) Which of the following option represents the fail function (or prefix function) for pattern ABABACA in Knuth-Morris-Pratt (KMP) algorithm?

- [0, 1, 1, 2, 3, 0, 1]
- [0, 0, 1, 2, 3, 0, 1]
- [0, 0, 1, 2, 3, 1, 1]
- [0, 0, 1, 2, 3, 0, 0]

No, the answer is incorrect.

Score: 0

Feedback:

In output list L of fail function, L[i] will represent the length of the longest proper prefix of W, which is also a proper suffix of W (considering W till the i th index only). So option (b) is the correct answer.

Accepted Answers:

[0, 0, 1, 2, 3, 0, 1]

2) For searching the pattern **aab** by Brute force algorithm on which of the following text will result in worst case behavior?

1 point

- abbaabbaabba
- abbbbbbbbbb
- babababababa
- aaaaaaaaaaaa

No, the answer is incorrect.

Score: 0

Feedback:

In option (d), the algorithm will take maximum time because, after every three successful matches, 4th character will mismatch then we will shift pattern one position, so, $9 * 4 = 36$ steps are required which is maximum in all options. Hence, option (d) is the correct answer.

Accepted Answers:

aaaaaaaaaaaa

3) We want to search a pattern in the text using Boyer-Moore skipping heuristic. For which of the following patterns the **last** dictionary (as discussed in the lectures) will be the biggest in size?

1 point

- bookkeeper
- boondoggle
- hypothetic
- automadata

No, the answer is incorrect.

Score: 0

Feedback:

For option (a) last will be: {b: 0, 'o': 2, 'k': 4, 'e': 8, 'p': 7, 'r': 9}, size = 6

For option (b) last will be: {b: 0, 'o': 5, 'n': 3, 'd': 4, 'g': 7, 'l': 8, 'e': 9}, size = 7

For option (c) last will be: {t: 5, 'y': 1, 'p': 2, 'o': 3, 't': 7, 'e': 6, 'l': 8, 'c': 9}, size = 8

For option (d) last will be: {a: 11, 'u': 1, 'v': 10, 'o': 3, 'm': 4, 'd': 8}, size = 6

Hence, option (c) dictionary has a maximum size of 8

Accepted Answers:

hypothetic

4) Which of the following is True with respect to the Rabin-Karp string matching algorithm?

1 point

- If the hash value of a substring of the text matches that of the pattern, then we can guarantee that the pattern has been found in the text.
- If the hash value of a substring of the text matches that of the pattern, then we have to match the actual substring to the pattern to guarantee that the pattern has been found.
- Rabin Karp algorithm and brute force pattern searching algorithm have the same worst case time complexity.
- Rabin Karp algorithm and Boyer Moore pattern searching algorithm have the same worst case time complexity

No, the answer is incorrect.

Score: 0

Feedback:

The hash function can return the same hash value for two different keys because of that option(a) is not correct. Other options are correct regarding the Rabin-Karp algorithm.

Accepted Answers:

If the hash value of a substring of the text matches that of the pattern, then we have to match the actual substring to the pattern to guarantee that the pattern has been found.
Rabin Karp algorithm and brute force pattern searching algorithm have the same worst case time complexity.
Rabin Karp algorithm and Boyer Moore pattern searching algorithm have the same worst case time complexity

5) Consider the below function that implements Boyer-Moore skipping heuristic to find all occurrences of pattern p in text t. Choose the correct option that can replace I (line #17) and II (line #20) marked in **1 point** the function below.

```
1 | def boyerMoore(t, p):  
2 |     matchList=[]  
3 |     n = len(t)  
4 |     m = len(p)  
5 |     if (m>n):  
6 |         return None  
7 |     match = True  
8 |     last = {p[i]:i for i in range(m)}  
9 |     i = m-1  
10 |    while i < n:  
11 |        match = True  
12 |        for j in range(m-1,-1,-1):  
13 |            c = t[i+j:m]  
14 |            if (c != p[j]):  
15 |                if (c in last):  
16 |                    # If character is in the pattern  
17 |                    i = last[c]  
18 |                else:  
19 |                    # If character is not in the pattern  
20 |                    i = II  
21 |                match = False  
22 |            break  
23 |        if (match):  
24 |            matchList.append(i-m+1)  
25 |            i += 1  
26 |  
27 |    if (len(matchList)>0):  
28 |        return matchList  
29 |    else:  
30 |        return None
```

- I = i+m and II = i+m-last[c]
- I = i+m-last[c] and II = i+m
- I = i+m-last[c]-1 and II = i+m
- I = i+m and II = i+m-last[c]-1

No, the answer is incorrect.

Score: 0

Feedback:

Option (c) is the correct code statement to fill to complete the Boyer-Moore skipping heuristic code.

Accepted Answers:

I = i+m-last[c]-1 and II = i+m

6) Consider the Rabin-Karp algorithm using modulo arithmetic to match pattern in base 10 . Taking modulo q=11 , how many **false positives** matches does the Rabin-Karp matcher encounter while searching pattern 36 in the text 591464256 ?

1 point

- 1
- 2
- 3
- None

No, the answer is incorrect.

Score: 0

Feedback:

In option (c), the algorithm takes 15 comparisons of characters, which is the maximum in all options because of that option (c) is correct.

Accepted Answers:

3

7) We want to search the pattern **strawberry** in the text **straw plus berry is strawberry** what is the count of character comparisons using Boyer-Moore skipping heuristic and Brute force pattern matching **1 point** respectively?

- 14, 37
- 13, 36
- 13, 37
- 14, 36

No, the answer is incorrect.

Score: 0

Feedback:

Option (a) values are representing the correct numbers of comparison of characters.

Accepted Answers:

14, 37

8) Which of the following combination of input text T and pattern P will exhibit the worst case running time behavior for Boyer-Moore skipping heuristic?

1 point

- T = 'bababababaabaa' and P = 'abb'
- T = 'aaaaaaaaaaaaaaaa' and P = 'abb'
- T = 'aaaaaaaaaaaaaaaa' and P = 'baa'
- T = 'aaaaaaaaaaaaaaaa' and P = 'bbb'

No, the answer is incorrect.

Score: 0

Feedback:

In option (c), the last state will be the final state q2 after completion of the string, because of that these strings will be accepted by a given finite automaton. In options (a) and (c), the last state will be after completion of string, which is not the final state because of that, these strings will not be accepted by given finite automaton.

Accepted Answers:

3

9) Consider the following finite automaton where q0 is the starting state and q2 is the final or accepting state , and input symbols are $\Sigma = \{a, b, c\}$.

1 point

Which of the following string(s) will be accepted by the given finite automaton?

- abbbabba
- aaaaaaaaa
- ababababab
- aabbaabccbbaab
- bcbcabcabcabcbaaa

No, the answer is incorrect.

Score: 0

Feedback:

In pattern (ba)* = {, 'ba', 'bab', 'babab', ...} so we can replace (ba)* by any value of set. So option (a), (b) and (d) are correct string will match with pattern.

Accepted Answers:

abbbabba, aaaaaaaaa, ababababab, aabbaabccbbaab, bcbcabcabcabcbaaa

10) Which of the following string(s) match the pattern ^ab(ba)*cd\$?

1 point

- abbacd
- abcd
- abbababa
- abbababacd
- abababacd

No, the answer is incorrect.

Score: 0

Feedback:

In pattern (ba)* = {, 'ba', 'bab', 'babab', ...} so we can replace (ba)* by any value of set. So option (a), (b) and (d) are correct string will match with pattern.

Accepted Answers:

abbacd, abcd, abbababacd

The due date for submitting this assignment has passed.

Due on 2022-07-17, 23:59 IST.

You may submit any number of times before the due date. The final submission will be considered for grading.

1) Solve the following linear programming problem by graphical method. Find out the correct vertex that minimize the objective function?

1 point

Minimize $Z = 20x + 50y$ subject to constraints:

- $x + 2y \geq 10$
 $3x + 4y \leq 24$
 $x \geq 0, y \geq 0$
- (8,0)
 - (0,6)
 - (4,3)
 - (10,0)

No, the answer is incorrect.

Score: 0

Feedback:

(8,0) and (10,0) can not be the correct vertex because these points are not satisfying the constraints. Other than that (0,6), (0,5), (1,5), (2,4) and (4,3) are possible vertices for given constraints out of these (4,3) minimized the objective function ' $Z = 20x + 50y$ ' with value 230.

Accepted Answers:

(4,3)

2) A plant manufacture produces two types of products A and B and sells them at a profit of Rs. 5 per item on type A and Rs. 3 per item on type B. Each product is processed on two machines G and H. One item of type A requires one minute of processing time on G and two minutes on H; One item of type B requires one minute on G and one minute on H. Machine G is available for not more than 5 hours 40 minutes, while machine H is available for 7 hours 20 minutes during any working day.

Find the formulation of LPP to Maximize the total profit?

Let X_1 be the number of item produced of type A and X_2 be the number of item produced of type B.

Maximize: $Z = 5X_1 + 3X_2$

- $X_1 + X_2 \leq 340$

$2X_1 + X_2 \leq 440$

$X_1 \geq 0, X_2 \geq 0$

Let X_1 be the number of products of type A and X_2 be the number of products of type B.

Maximize: $Z = 5X_1 + 3X_2$

- $X_1 + 2X_2 \leq 340$

$X_1 + X_2 \leq 440$

$X_1 \geq 0, X_2 \geq 0$

Let X_1 be the number of products of type A and X_2 be the number of products of type B.

Maximize: $Z = 340X_1 + 440X_2$

- $X_1 + 2X_2 \leq 5$

$X_1 + X_2 \leq 3$

$X_1 \geq 0, X_2 \geq 0$

No, the answer is incorrect.

Score: 0

Feedback:

Let X_1 be the number of items produced of type A and X_2 be the number of items produced of type B.

* Profit on A is 5 and profit on B is 3 then the objective function will be 'Maximize': $Z = 5X_1 + 3X_2$

* Machine G is available 340 minutes, and it consumes 1 minute on A and 1 minute on B then the first constraint will be $X_1 + X_2 \leq 340$.

* Machine H is available 440 minutes, and it consumes 2 minutes on A and 1 minute on B then the second constraint will be $2X_1 + X_2 \leq 440$.

* And the quantity of A and B can not be negative then the third constraint will be $X_1 \geq 0, X_2 \geq 0$

Accepted Answers:

Let X_1 be the number of item produced of type A and X_2 be the number of item produced of type B.

Maximize: $Z = 5X_1 + 3X_2$

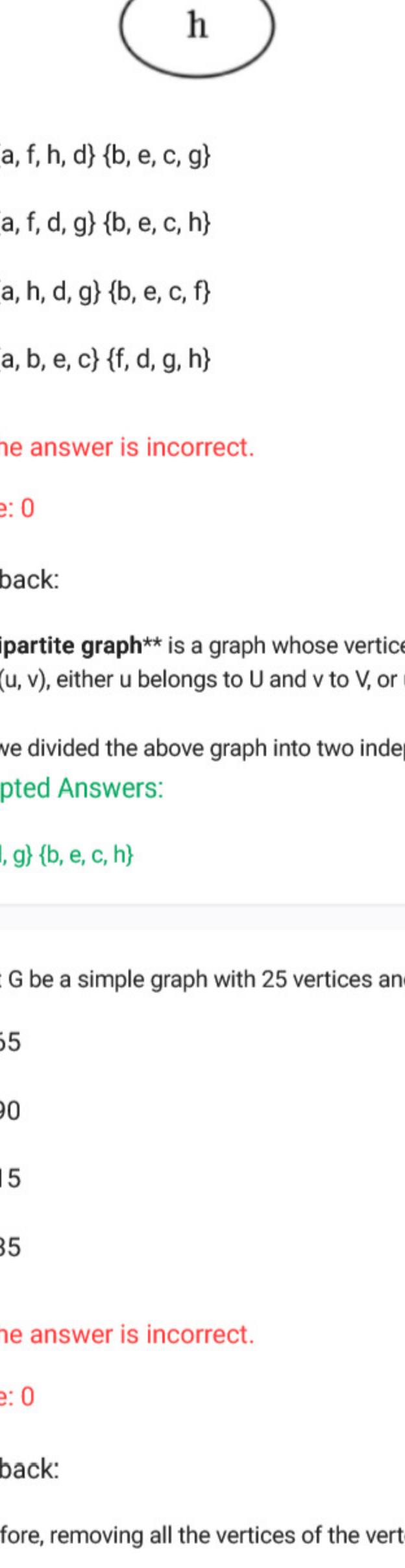
$X_1 + X_2 \leq 340$

$2X_1 + X_2 \leq 440$

$X_1 \geq 0, X_2 \geq 0$

3) Find the max-flow in the following transport network, where a is the source and z is the destination?

1 point



- 15
- 13
- 11
- 16

No, the answer is incorrect.

Score: 0

Feedback:

For example, if we select the following order of paths to calculate/update residual capacity in the transport network, then we will get the maximum capacity of 13.

'a - c - z' bottleneck capacity/flow = 3

'a - b - d - z' bottleneck capacity/flow = 4

'a - c - e - z' bottleneck capacity/flow = 5

'a - b - e - z' bottleneck capacity/flow = 1

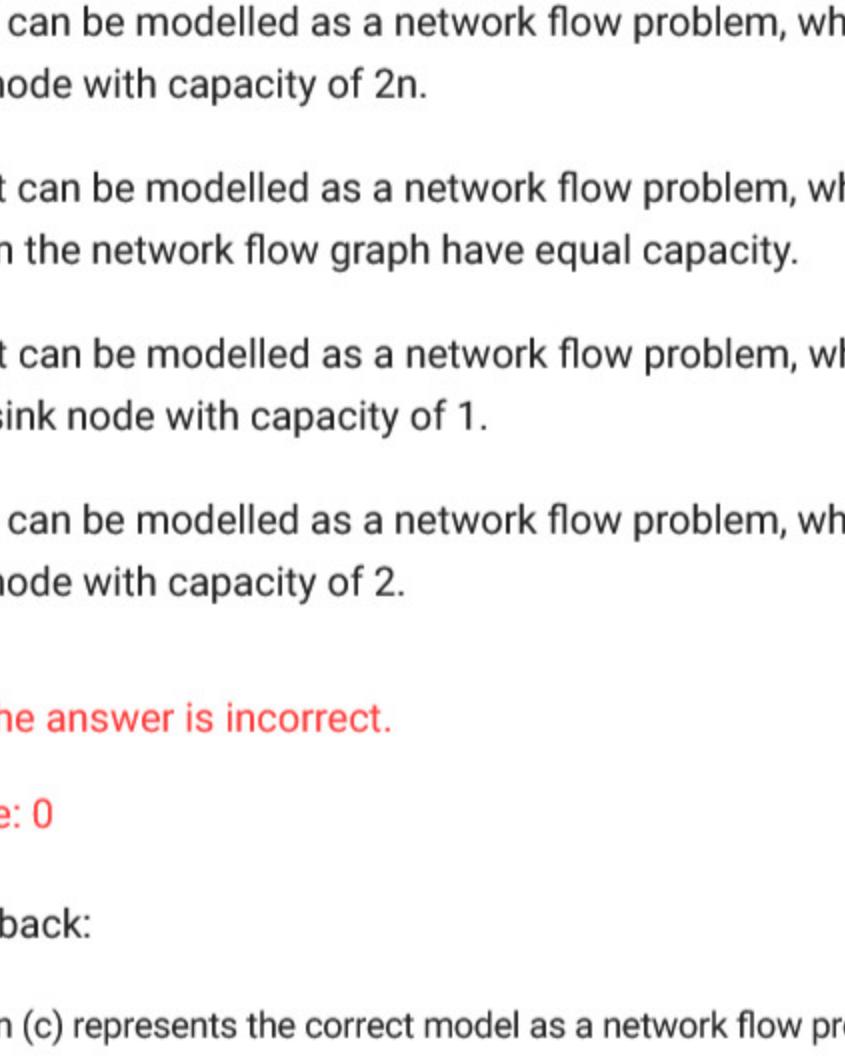
After that, there is no path available for flow to 'z' then the sum of all above flow is 13.

Accepted Answers:

((b,d),(e,z),(c,z))

5) Which of the following set of vertices can make the graph Bipartite?

1 point



- {a, f, h, d} {b, e, c, g}
- {a, f, d, g} {b, e, c, h}
- {a, h, d, g} {b, e, c, f}
- {a, b, e, c} {f, d, g, h}

No, the answer is incorrect.

Score: 0

Feedback:

Therefore, removing all the vertices of the vertex cover from the graph results in an isolated graph and the same set of nodes would be the independent set in the original graph.

Number of independent set = number of total vertices - the size of the minimum vertex cover

Number of independent set = $25 - 10 = 15$

Accepted Answers:

((b,d),(e,z),(c,z))

6) Let G be a simple graph with 25 vertices and 50 edges. The size of the minimum vertex cover of G is 10. What is the size of the maximum independent set of G ?

1 point

- 65
- 90
- 15
- 35

No, the answer is incorrect.

Score: 0

Feedback:

For every, if we select the following order of paths to calculate/update residual capacity in the transport network, then we will get the maximum capacity of 13.

'a - c - z' bottleneck capacity/flow = 3

'a - b - d - z' bottleneck capacity/flow = 4

'a - c - e - z' bottleneck capacity/flow = 5

'a - b - e - z' bottleneck capacity/flow = 1

After that, there is no path available for flow to 'z' then the sum of all above flow is 13.

Accepted Answers:

((b,d),(e,z),(c,z))

7) Two CNF formulas are given. Choose the correct statement ?

1 point

(I) $(a \vee \neg b \vee \neg c) \wedge (\neg a \vee \neg b \vee \neg c) \wedge (b \vee c \vee \neg a)$

(II) $(a \vee \neg b \vee c) \wedge (\neg a \vee \neg b \vee c) \wedge (\neg a \vee b \vee c) \wedge (\neg a \vee b \vee \neg c) \wedge (a \vee b \vee c) \wedge (a \vee b \vee \neg c)$

(III) (I) is unsatisfiable (II) is satisfiable

(IV) (I) is satisfiable (II) is unsatisfiable

(V) (I) is satisfiable (II) is satisfiable

No, the answer is incorrect.

Score: 0

Feedback:

CNF formula (I) return 'True' or 'False' for some combination value of 'a', 'b' and 'c' but CNF formula (II) always returns False for all combination value of a, b and c. Hence, option (C) is correct.

Accepted Answers:

((b,d),(e,z),(c,z))

8) In the Ford-Fulkerson algorithm to find the maximum flow, what do all edges in the residual graph having value 0 denote?

1 point

Edges with flow equal to maximum capacity

Edges with no flow

Edges with partial flow

Edges with flow in opposite direction

No, the answer is incorrect.

Score: 0

Feedback:

If all edges in the residual graph have a value of 0 then it denotes that the flow on the edge is equal to maximum capacity.

Accepted Answers:

Edges with flow equal to maximum capacity

9) Assume there are n teachers and 2n subjects. Each teacher has to teach exactly 2 subjects. Their preferences are modelled as a directed graph G, such that there exists an edge from a teacher node T_i to a subject node S_j in G, if T_i prefers teaching S_j . How can this problem be modelled as a network flow problem?

1 point

t can be modelled as a network flow problem, where the source node is connected to every teacher node in G with capacity of n, and every subject node in G is connected to the sink node with capacity of 2n.

It can be modelled as a network flow problem, where the source node is connected to every teacher node in G with capacity of 2, and every subject node in G is connected to the sink node with capacity of 1.

t can be modelled as a network flow problem, where the source node is connected to every teacher node in G with capacity of 1, and every subject node in G is connected to the sink node with capacity of 2.

No, the answer is incorrect.

Score: 0

Feedback:

Option (C) represents the correct model as a network flow problem for a given scenario in the problem statement.

Accepted Answers:

It can be modelled as a network flow problem, where the source node is connected to every teacher node in G with capacity of 2, and every subject node in G is connected to the sink node with capacity of 1.

10) Let C be a problem that belongs to the class NP. Then which one of the following is TRUE?

1 point

(I) If C is NP-Hard, then it is NP-complete.

(II) There is no polynomial time algorithm for C.

(III) If C can be solved in polynomial time, then P = NP.

(IV) If every problem in NP is reducible to C in polynomial time then C is NP-complete.

No, the answer is incorrect.

Score: 0

Feedback:

Option (B) is incorrect because set NP includes both P(Polynomial-time solvable) and NP-Complete. Option (C) is incorrect because C may belong to P (same reason as (B)). Other options (A) and (D) are correct.

Accepted Answers:

If C is NP-Hard, then it is NP-complete.

If every problem in NP is reducible to C in polynomial time then C is NP-complete.