
Started on Friday, 9 May 2025, 8:21 AM

State Finished

Completed on Tuesday, 13 May 2025, 9:43 PM

Time taken 4 days 13 hours

Overdue 4 days 11 hours

Grade 80.00 out of 100.00

Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to implement merge sort using iterative approach on the given list of float values.

For example:

Test	Input	Result
Merge_Sort(S)	5 10.2 21.3 3.5 7.8 9.8	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]
Merge_Sort(S)	6 20.3 41.2 5.3 6.2 8.1 65.2	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]

Answer: (penalty regime: 0 %)

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
def Merge_Sort(S):
    if(len(S)>1):
        mid = len(S)//2
        left = S[:mid]
        right = S[mid:]
        Merge_Sort(left)
        Merge_Sort(right)
        i = j = k = 0
        while(i < len(left) and j < len(right)):
            if(left[i] < right[j]):
                S[k] = left[i]
                i = i +1
            else:
                S[k] = right[j]
                j = j+1
            k = k+1
        while(i<len(left)):
            S[k] = left[i]
```

	Test	Input	Expected	Got	
✓	Merge_Sort(S)	5 10.2 21.3 3.5 7.8 9.8	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]	The Original array is: [10.2, 21.3, 3.5, 7.8, 9.8] Array after sorting is: [3.5, 7.8, 9.8, 10.2, 21.3]	✓
✓	Merge_Sort(S)	6 20.3 41.2 5.3 6.2 8.1 65.2	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]	The Original array is: [20.3, 41.2, 5.3, 6.2, 8.1, 65.2] Array after sorting is: [5.3, 6.2, 8.1, 20.3, 41.2, 65.2]	✓

	Test	Input	Expected	Got	
✓	Merge_Sort(S)	4 2.3 6.1 4.5 96.5	The Original array is: [2.3, 6.1, 4.5, 96.5] Array after sorting is: [2.3, 4.5, 6.1, 96.5]	The Original array is: [2.3, 6.1, 4.5, 96.5] Array after sorting is: [2.3, 4.5, 6.1, 96.5]	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Create a Dynamic Programming python Implementation of Coin Change Problem.

For example:

Test	Input	Result
count(arr, m, n)	3 4 1 2 3	4

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
def count(S, m, n):
    table = [[0 for x in range(m)] for x in range(n+1)]
    for i in range(m):
        table[0][i] = 1
    for i in range(1, n+1):
        for j in range(m):

            x = table[i - S[j]][j] if i-S[j] >= 0 else 0

            y = table[i][j-1] if j >= 1 else 0

            # total count
            table[i][j] = x + y

    return table[n][m-1]
```

	Test	Input	Expected	Got	
✓	count(arr, m, n)	3 4 1 2 3	4	4	✓
✓	count(arr, m, n)	3 16 1 2 5	20	20	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Write a Python program to Implement Minimum cost path in a Directed Graph

For example:

Test	Result
getMinPathSum(graph, visited, necessary, source, dest, 0);	12

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
minSum = 1000000000
def getMinPathSum(graph, visited, necessary,
                  src, dest, currSum):

    global minSum
    if (src == dest):
        flag = True;
        for i in necessary:
            if (not visited[i]):
                flag = False;
                break;
        if (flag):
            minSum = min(minSum, currSum);
        return;

    else:
        visited[src] = True;
```

	Test	Expected	Got	
✓	getMinPathSum(graph, visited, necessary, source, dest, 0);	12	12	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **4**

Not answered

Mark 0.00 out of 20.00

Write a python program to find the maximum contiguous subarray on the given float array using kadane's algorithm.

For example:

Test	Input	Result
s.maxSubArray(A)	5 -9.6 -3.5 6.3 8.31 9.2	The sum of contiguous sublist with the largest sum is 23.8

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
def maxSubArray(a,size):  
  
    ##### Add your Code here #####  
  
A=[]  
n=int(input())  
for i in range(n):  
    A.append(float(input()))  
s=Solution()  
print("The sum of contiguous sublist with the largest sum is {:.1f}".format(s.maxSubArray(A)))
```

Question **5**

Correct

Mark 20.00 out of 20.00

Create a python program to find Minimum number of jumps to reach end of the array using naive method(recursion) using float values

For example:

Test	Input	Result
minJumps(arr, 0, n-1)	6 2.3 7.4 6.3 1.5 8.2 0.1	Minimum number of jumps to reach end is 2

Answer: (penalty regime: 0 %)

Reset answer

Ace editor not ready. Perhaps reload page?

Falling back to raw text area.

```
def minJumps(arr, l, h):
    if (h == l):
        return 0
    if (arr[l] == 0):
        return float('inf')
    min = float('inf')
    for i in range(l + 1, h + 1):
        if (i < l + arr[l] + 1):
            jumps = minJumps(arr, i, h)
            if (jumps != float('inf') and
                jumps + 1 < min):
                min = jumps + 1

    return min
arr = []
n = int(input())
for i in range(n):
    arr.append(float(input()))
```

	Test	Input	Expected	Got	
✓	minJumps(arr, 0, n-1)	6 2.3 7.4 6.3 1.5 8.2 0.1	Minimum number of jumps to reach end is 2	Minimum number of jumps to reach end is 2	✓
✓	minJumps(arr, 0, n-1)	10 3.2 3.2 5 6.2 4.9 1.2 5.0 7.3 4.6 6.2	Minimum number of jumps to reach end is 2	Minimum number of jumps to reach end is 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.