

# TT DS PYTHON MODULE-23

**Started on** Thursday, 3 October 2024, 11:13 AM**State** Finished**Completed on** Thursday, 3 October 2024, 11:20 AM**Time taken** 7 mins 33 secs**Grade** 80.00 out of 100.00

Question 1

Not answered

Mark 0.00 out of 20.00

Flag question

Write a recursive python function to perform merge sort on the unsorted list of float values.

**For example:**

Test	Input	Result
mergesort(li)	5 3.2 1.5 1.6 1.7 8.9	[1.5, 1.6, 1.7, 3.2, 8.9]
mergesort(li)	6 3.1 2.3 6.5 4.5 7.8 9.2	[2.3, 3.1, 4.5, 6.5, 7.8, 9.2]

**Answer:** (penalty regime: 0 %)

1

Syntax Error(s)

File "\_\_tester\_\_.python3", line 1

Write a python program to Implement Minimum cost path using Dynamic Programming.

^

SyntaxError: invalid syntax

Incorrect

Marks for this submission: 0.00/20.00.

Question 2

Correct

Mark 20.00 out of 20.00

Flag question

**Print All Paths With Minimum Jumps**

1. You are given a number N representing number of elements.
  2. You are given N space separated numbers (ELE : elements).
  3. Your task is to find & print
    - 3.1) "MINIMUM JUMPS" need from 0th step to (n-1)th step.
    - 3.2) all configurations of "MINIMUM JUMPS".
- NOTE: Checkout sample question/solution video inorder to have more insight.

**For example:**

Test	Input	Result
minJumps(arr)	10 3 3 0 2 1 2 4	0 -> 3 -> 5 -> 6 -> 9 0 -> 3 -> 5 -> 7 -> 9

	2	
	0	
	0	

**Answer:** (penalty regime: 0 %)

Reset answer

```

1 from queue import Queue
2 import sys
3 class Pair(object):
4     idx = 0
5     psf = ""
6     jmps = 0
7     def __init__(self, idx, psf, jmps):
8
9         self.idx = idx
10        self.psf = psf
11        self.jmps = jmps
12 def minJumps(arr):
13     ##### Add your Code here.
14     #Start here
15     MAX_VALUE = sys.maxsize
16     dp = [MAX_VALUE for i in range(len(arr))]
17     n = len(dp)
18     dp[n - 1] = 0
19     for i in range(n - 2, -1, -1):
20         steps = arr[i]
21         minimum = MAX_VALUE
22         for j in range(1, steps + 1, 1):

```

	Test	Input	Expected	Got	
	minJumps(arr)	10 3 3 0 2 1 2 4 2 0 0	0 -> 3 -> 5 -> 6 -> 9 0 -> 3 -> 5 -> 7 -> 9	0 -> 3 -> 5 -> 6 -> 9 0 -> 3 -> 5 -> 7 -> 9	
	minJumps(arr)	7 5 5 0 3 2 3 6	0 -> 1 -> 6 0 -> 3 -> 6 0 -> 4 -> 6 0 -> 5 -> 6	0 -> 1 -> 6 0 -> 3 -> 6 0 -> 4 -> 6 0 -> 5 -> 6	

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Correct

Mark 20.00 out of 20.00

Flag question

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

```

1 minSum = 1000000000
2 def getMinPathSum(graph, visited, necessary,
3     src, dest, currSum):
4
5     ##### Add your Code here #####
6     global minSum
7     if (src == dest):
8         flag = True;
9         for i in necessary:
10            if (not visited[i]):
11                flag = False;
12            break;

```

```

13         if (flag):
14             minSum = min(minSum, currSum);
15         return;
16
17     else:
18         visited[src] = True;
19         for node in graph[src]:
20             if not visited[node[0]]:
21                 visited[node[0]] = True;
22                 getMinPathSum(graph, visited,

```

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

Correct

Mark 20.00 out of 20.00

Flag question

Create a Python Function to find the total number of distinct ways to get a change of 'target' from an unlimited supply of coins in set 'S'.

For example:

Test	Input	Result
count(S, len(S) - 1, target)	3 4 1 2 3	The total number of ways to get the desired change is 4

Answer: (penalty regime: 0 %)

Reset answer

```

1 def count(S, n, target):
2     ##### Add Your Code Here #####
3     #Start here
4     if target == 0:
5         return 1
6     if target < 0 or n < 0:
7         return 0
8     incl = count(S, n, target - S[n])
9     excl = count(S, n - 1, target)
10    return incl + excl
11    #End here
12
13 if __name__ == '__main__':
14     S = [1, 2, 3]
15     n=int(input())
16     target = int(input())
17     for i in range(n):
18         S.append(int(input()))
19     print('The total number of ways to get the desired change is',
20         count(S, len(S) - 1, target))

```

Test	Input	Expected	Got
count(S, len(S) - 1, target)	3 4 1 2 3	The total number of ways to get the desired change is 4	The total number of
count(S, len(S) - 1, target)	3 11 1 2 5	The total number of ways to get the desired change is 11	The total number of

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Question 5

Correct

Mark 20.00 out of 20.00

Flag question

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

```

1 class Solution:
2     ##### Add your Code here #####
3     def maxSubArray(self,A):
4         res=0
5         mm= -10000
6         for v in A:
7             res+=v
8             mm=max(mm,res)
9             if res<0:
10                res=0
11         return mm
12 A=[]
13 n=int(input())
14 for i in range(n):
15     A.append(float(input()))
16 s=Solution()
17 print("The sum of contiguous sublist with the largest sum is {:.1f}".format(s.maxSubArray(A)))

```

Test	Input	Expected	Got
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	The sum of contiguous sublist
s.maxSubArray(A)	7 2.3 6.5 4.6 -7.8 -2.8 -1.6 9.8	The sum of contiguous sublist with the largest sum is 13.4	The sum of contiguous sublist

Passed all tests!

Correct

Marks for this submission: 20.00/20.00.

Finish re