

3.10 COUNTING TUPLES WITH ZERO SUM FROM FOUR LISTS

Question:

Given four lists A, B, C, D of integer values, Write a program to compute how many tuples $n(i, j, k, l)$ there are such that $A[i] + B[j] + C[k] + D[l]$ is zero.

AIM

To design a Python program that efficiently counts the number of tuples from four lists whose elements sum to zero.

ALGORITHM

1. Use a hash map to store the frequency of all possible sums of elements from A and B.
2. For each possible sum of elements from C and D, check if the negated sum exists in the hash map.
3. If it does, add the frequency to the result count.

PROGRAM

```
from collections import Counter

def four_sum_count(A, B, C, D):
    AB = Counter(a + b for a in A for b in B)
    return sum(AB[-(c + d)] for c in C for d in D)

def run_four_sum():
    A = list(map(int, input("Enter list A: ").split()))
    B = list(map(int, input("Enter list B: ").split()))
    C = list(map(int, input("Enter list C: ").split()))
    D = list(map(int, input("Enter list D: ").split()))
    print("Tuples count:", four_sum_count(A, B, C, D))

run_four_sum()
```

Input:

$(-1, 2), (2, 1), (0, 2), (-1, -2)$

Output:

```
>>> Enter list A: -1 2
      Enter list B: 2 1
      Enter list C: 0 2
      Enter list D: -1 -2
      Tuples count: 2
```

RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: $O(n^2)$, where n is the length of each list.
- Space Complexity: $O(n^2)$ for storing pairwise sums.