Matrix Diagonal Sum – Complete Guide

Problem:
Given a square matrix, find the sum of its primary and secondary diagonals. If the matrix size is odd, count the center element only once.

Example:
Matrix:
1 2 3
4 5 6
7 8 9
Output: 25

Approach 1: Brute Force O(n²)
Traverse the full matrix and add elements where i==j or i+j==n-1. Subtract center once if n is odd.

Approach 2: Optimized O(n)
Traverse once. Add mat[i][i] and mat[i][n-i-1]. If both indices are same, add only once.

Java Code (O(n)):
```
class Solution {
  public int diagonalSum(int[][] mat) {
    int n = mat.length, sum = 0;
    for(int i=0;i       sum += mat[i][i];
      if(i != n-i-1) sum += mat[i][n-i-1];
    }
    return sum;
  }
}
```

Python Code (O(n)):
```
def diagonalSum(mat):
  n = len(mat)
  total = 0
  for i in range(n):
    total += mat[i][i]
    if i != n-i-1:
      total += mat[i][n-i-1]
  return total
```

C++ Code (O(n)):
```
class Solution {
public:
  int diagonalSum(vector>& mat) {
    int n = mat.size(), sum = 0;
    for(int i=0;i       sum += mat[i][i];
      if(i != n-i-1) sum += mat[i][n-i-1];
    }
    return sum;
  }
};
```

Complexity:
Brute Force: O(n²), Optimized: O(n), Space: O(1)