Minimum Changes to Form 'Y' in an n x n Matrix

Problem Statement:

Given an odd-sized square matrix (n x n) that contains only 0s, 1s, and 2s, your task is to find the minimum number of changes needed to form the letter 'Y' on the matrix.

The letter 'Y' is formed when:
- The two diagonals from the upper left and upper right corners meet at the center of the matrix.
- A vertical line stretches downward from the center.
- All numbers forming the 'Y' must be the same.
- The background numbers (not part of the 'Y') must also be the same but different from the 'Y' numbers.

There are exactly 6 possible ways to form 'Y' using different pairs of numbers (0,1), (0,2), (1,0), etc.

Python Solution:

```
def min_changes_to_form_y(matrix):
    n = len(matrix)
    possible_values = [0, 1, 2]
    min_changes = float('inf')

    y_indices = set()

    for i in range(n // 2 + 1):
        y_indices.add((i, i))
        y_indices.add((i, n - 1 - i))

    for i in range(n // 2, n):
        y_indices.add((i, n // 2))

    for y_val in possible_values:
        for bg_val in possible_values:
            if y_val == bg_val:
                continue

            changes = 0
            for i in range(n):
                for j in range(n):
                    if (i, j) in y_indices:
                        if matrix[i][j] != y_val:
                            changes += 1
                    else:
                        if matrix[i][j] != bg_val:
                            changes += 1
```

```
            min_changes = min(min_changes, changes)

    return min_changes
```

Explanation:

- Identify the 'Y' shape in the matrix using a set of coordinates.
- Iterate over all six possible (Y-value, background-value) combinations.
- Count how many changes are needed to transform the matrix into each pattern.
- Return the minimum number of changes across all patterns.

Time and Space Complexity:

- Time Complexity: $O(n^2)$
  - We iterate over each cell in the n x n matrix, making multiple passes.
- Space Complexity: $O(n^2)$
  - The set storing 'Y' indices takes $O(n^2)$ space in the worst case.