Birthday Gift - Editorial

Difficulty:

Medium - Hard

Prerequisites:

Strings
Hashing - (Tutorial)
Manacher's Algorithm - (Tutorial)
Suffix Array - (Tutorial)
Binary Search - (Tutorial)

Problem in Brief:

You are given a string S. You have to count the number of odd length substrings of S that are also palindromes.

Editorial:

There are 3 ways to solve this problem :-

1) Using Hashing
2) Using Manacher's Algorithm
3) Using Suffix Array

I will explain the Suffix Array algorithm as it is most intuitive out of all 3 ways.

Birthday Gift - Editorial

Consider an odd length palindromic substring of S. It must have it's center at one of the indices of S. Instead of counting the substrings ad-hoc we will count the odd length palindromic substrings with its center at index i for all 1 <= i <= |S|.
The count of odd length palindromic substrings with center at i is equal to precisely (MAX(i) + 1) / 2.
Where MAX(i) = length of longest odd length palindromic substring with its center at i.

To find MAX(i) we will build the Suffix Array on the concatenation of S and reverse of S.
We will then use binary search to find the max odd length substring using Longest Common Prefix from the Suffix Array DS.

We will then sum (MAX(i) + 1) / 2 for 1 <= i <= |S|
And output the answer.

Time Complexity:

Building the suffix array is the costliest operation. Binary search will only take log(N) time per index. Hence the Time Complexity is

O(N * log^2(N))

Similar Problems:

First
Second
Third