Write a function to check whether an input string is a valid IPv4 address or IPv6 address or neither.

**IPv4** addresses are canonically represented in dot-decimal notation, which consists of four decimal numbers, each ranging from 0 to 255, separated by dots ("."), e.g.,172.16.254.1;

Besides, leading zeros in the IPv4 is invalid. For example, the address 172.16.254.01 is invalid.

**IPv6** addresses are represented as eight groups of four hexadecimal digits, each group representing 16 bits. The groups are separated by colons (":"). For example, the address 2001:0db8:85a3:0000:0000:8a2e:0370:7334 is a valid one. Also, we could omit some leading zeros among four hexadecimal digits and some low-case characters in the address to upper-case ones, so 2001:db8:85a3:0:0:8A2E:0370:7334 is also a valid IPv6 address(Omit leading zeros and using upper cases).

However, we don't replace a consecutive group of zero value with a single empty group using two consecutive colons (::) to pursue simplicity. For example, 2001:0db8:85a3::8A2E:0370:7334 is an invalid IPv6 address.

Besides, extra leading zeros in the IPv6 is also invalid. For example, the address 02001:0db8:85a3:0000:0000:8a2e:0370:7334 is invalid.

**Note:** You may assume there is no extra space or special characters in the input string.

**Example 1:**

**Input:** "172.16.254.1"

**Output:** "IPv4"

**Explanation:** This is a valid IPv4 address, return "IPv4".

**Example 2:**

**Input:** "2001:0db8:85a3:0:0:8A2E:0370:7334"

**Output:** "IPv6"

**Explanation:** This is a valid IPv6 address, return "IPv6".

**Example 3:**

**Input:** "256.256.256.256"

**Output:** "Neither"

**Explanation:** This is neither a IPv4 address nor a IPv6 address.