468. Validate IP Address

Add to List

Description (?tab=Description) Submission (?tab=Submission) Solutions (?tab=Solutions)

Total Accepted: **3572** Total Submissions: **17405** Difficulty: **Medium** Contributors: **Cyber233** (/cyber233/)

Notes

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.
```

Hide Company Tags Twitter (/company/twitter/)

Hide Tags String (/tag/string/)

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/598) Pick One (/problems/random-one-question/) Editorial Solution

C++

```
class Solution {
public:

    bool validIp4(string& ip) {
        if(ip.size() > 15) return false;
        int i = 0, j = 0;
        int region = 0;
        while(j <= ip.size()) {
```

✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
            if(j != ip.size() && ip[j] >= '0' && ip[j] <= '9') {
                j++;
            } else if(j == ip.size() || ip[j] == '.') {
                int len = j - i;
                if(len <= 0) return false;
                string t = ip.substr(i, len);
                int val = stoi(t);
                if((val != 0 && t[0] == '0') || val < 0 || val > 255 || (val == 0 && t != "0")) {
                    return false;
                }
                i = ++j;
                region++;
            } else {
                return false;
            }
        }
        return region == 4;
    }

    bool validIp6(string& ip) {
        if(ip.size() > 39) return false;
        int i = 0, j = 0, region = 0;
        while(j <= ip.size()) {
            if(j != ip.size() && (ip[j] >= '0' && ip[j] <= '9') || (ip[j] >= 'a' && ip[j] <= 'f') || (ip[j] >= 'A' && ip[j]
                'F')) {
                j++;
            } else if(j == ip.size() || ip[j] == ':') {
                int len = j - i;
                if(len <= 0 || len > 4) return false;
                i = ++j;
                region++;
            } else {
                return false;
            }
        }
        return region == 8;
    }


    string validIPAddress(string IP) {
        if(validIp4(IP)) return "IPv4";
        else if(validIp6(IP)) return "IPv6";
        else return "Neither";
    }
};
```

Notes

Custom Testcase □

Contribute Testcase

Run Code   Submit Solution

Frequently Asked Questions (/faq/) | Terms of Service (/tos/)

Privacy

✉ Send Feedback (mailto:admin@leetcode.com?subject=Feedback)