```
Const fn = (ip) => {
    if(isIPV4(ip)){
        return 'IPV4';
    }
    if( isIPV6(ip)){
        return 'IPV6';
    }
    return 'Neither';
}
```

test cases:

ipv4: strs.length === 4. 1 ≤ str.length ≤ 3

str no leading zero.

nam 0 ≤ num ≤ 255.

almost pass!

```
function isIPV4(ip) {
    const strs = ip.split('.');
    if(strs.length!==4) return false;
    for ( const str of strs) {
        if(str.length >= 2 && str[0] === '0') {
            return false;                 ||!( str.length >= 1 && str.length <= 3)
        }
    }                 => need to verify if consisting only 0-9.
    const nums = strs.map(str => parseInt(str));
    for(const num of nums) {
        if (!(num >= 0 && num <= 255)) {
            return false;
        }
    }
    return true;
}
```

```
function isIPv6(ip) {
    const strs = ip.split(':');
    if(strs.length !=== 8) {
        return false;
    }
    for(const str of strs) {
        if(!(str.length >=1 &&
                str.length <= 4)) {
            return false;
        }
        if(str.length >= 2 && str[0] === '0') {
            for(const c of str) {
                if(c !== '0') return false;
            }
        }
    }
    const nums = strs.map(str => parseInt(str));
    for(const num of nums) {
        If(!(num >= 0 && num <= 2** 16)) {
            return false;
        }
    }
    return true;
}
```

no need for leading zero check!

⇒ need to verify if consisting only valid hex numbers.

test cases:

8 groups.

② 16 bites ⇒ $0 \le num \le 2^{16}$

① $1 \le$ str.length ≤ 4.

if leading zero, num must be 0.

strs.length === 8.

A. validate str length

B. validate leading zero.