版本一：

Next[0] = -1

KMP 算法代码如下：

/\*\*

 \* @param {string} haystack

 \* @param {string} needle

 \* @return {number}

 \*/

var strStr = function(haystack, needle) {

    if(needle.length == 0)

    return 0;

    let len = needle.length,next = Array(len).fill(-1);

    let getNext = function(s)

    {

        if(len==0)

        return ;

        next[1] = 0;

        for(let i=2;i<s.length;i++)

        {

            let cur = next[i-1];

            while(cur!==-1 && s[i-1]!==s[cur])

            {

                cur = next[cur];

            }

            next[i] = cur+1;

        }

    }

    let kmp = function()

    {

        getNext(needle);

        //console.log(next);

        let i=0,j=0;

        while(i<haystack.length && j<len)

        {

            //console.log(i,j);

            if(j==-1 || haystack[i]==needle[j])

            {

                i++;

                j++;

            }

            else

            {

                j = next[j];

            }

        }

        //console.log(i,j);

        if(j==needle.length)

            return i-j;

        return -1;

    }

    return kmp();

};

版本二：

Next[0] =0;

/\*\*

 \* @param {string} haystack

 \* @param {string} needle

 \* @return {number}

 \*/

var strStr = function(haystack, needle) {

    if(needle.length == 0)

    return 0;

    let len = needle.length,next = Array(len).fill(0);

    let getNext = function(s)

    {

        if(len==0)

        return ;

        for(let i=1;i<s.length;i++)

        {

            let cur = next[i-1];

            while(cur>0 && s[i]!==s[cur])

            {

                cur = next[cur-1];

            }

            if(s[i]==s[cur])

                cur++;

            next[i] = cur;

        }

    }

    let kmp = function()

    {

        getNext(needle);

        //console.log(next);

        let i=0,j=0;

        while(i<haystack.length && j<len)

        {

            //console.log(i,j);

            if(haystack[i]==needle[j])

            {

                i++;

                j++;

            }

            else

            {

                if(j>0)

                    j = next[j-1];

                else

                    i++;

            }

        }

        //console.log(i,j);

        if(j==needle.length)

            return i-j;

        return -1;

    }

    return kmp();

};