

最大重复子字符串

给你一个字符串 `sequence`，如果字符串 `word` 连续重复 `k` 次形成的字符串是 `sequence` 的一个子字符串，那么单词 `word` 的 **重复值为 `k`**。单词 `word` 的 **最大重复值** 是单词 `word` 在 `sequence` 中最大的重复值。如果 `word` 不是 `sequence` 的子串，那么重复值 `k` 为 0。

给你一个字符串 `sequence` 和 `word`，请你返回 **最大重复值 `k`**。

示例 1:

输入: `sequence = "ababc", word = "ab"`

输出: 2

解释: "abab" 是 "ababc" 的子字符串。

示例 2:

输入: `sequence = "ababc", word = "ba"`

输出: 1

解释: "ba" 是 "ababc" 的子字符串，但 "baba" 不是 "ababc" 的子字符串。

示例 3:

输入: `sequence = "ababc", word = "ac"`

输出: 0

解释: "ac" 不是 "ababc" 的子字符串。

C 语言:

```
int maxRepeating(char * sequence, char * word){
    int l1=strlen(sequence),l2=strlen(word);
    int count=0,sum=0;
    int i=0;

    if((sequence==NULL) || (word==NULL))    return 0;
    if(l1<l2)    return 0;

    while(i<=l1-l2)
    {
        if(strncmp(&sequence[i],word,l2)==0)
        {
            count=count+1;
        }
    }
}
```

```

        i=i+12;//匹配成功，计数+1，偏移 word 长度继续匹配
    }
    else
    {
        count=0;
        i++;//匹配不成功，偏移一位继续匹配
    }
    if(sum<count) sum=count;
}
return sum;
}

```

C++: sequence 中找连续子串 注意 是连续

```

class Solution {
public:
    int maxRepeating(string sequence, string word) {
        string s=word;
        int cnt=0;
        while(sequence.find(s)!=string::npos){
            s+=word;
            cnt++;
        }
        return cnt;
    }
};

```