

找到给定字符串（由小写字母组成）中的最长子串 T ，要求 T 中的每一字符出现次数都不少于 k 。输出 T 的长度。

示例 1:

输入:

```
s = "aaabb", k = 3
```

输出:

3

最长子串为 "aaa"，其中 'a' 重复了 3 次。

示例 2:

输入:

```
s = "ababbc", k = 2
```

输出:

5

最长子串为 "ababb"，其中 'a' 重复了 2 次，'b' 重复了 3 次。

思路，分治法，把先动态的查找可能存在最长字符串的函数，然后递归调用子字符串，得到最长。

代码

```
class Solution {
public:
    int longestSubstring(string s, int k) {
        if(k<=0)
            return s.size();
        if(s.size()<k)
            return 0;
        unordered_map<char,int>work;
        for(auto i:s)
            work[i]++;
        bool flag=true;
        for(auto i:work){
            if(i.second<k)
                flag=false;
        }
        if(flag==true)
            return s.size();
        int pre=0;
        int maxLen=0;
        int i=0;
        for(;i<s.size();++i){
            if(work[s[i]]<k){
```

```
        maxlen=max(maxlen,longestSubstring(s.substr(pre,i-
pre),k));
        pre=i+1;
    }
}
maxlen=max(maxlen,longestSubstring(s.substr(pre,i-pre),k));
return maxlen;

}

};
```