

InfyTQ (Longest substring with unique chars)

August 7, 2019

0.1 InfyTQ Problem

Find longest substring of unique characters which is case insensitive

```
In [31]: mystr=input()
mysubstr=[]
mysubstr2={}
startindex=0
endindex=0
def subString(s):
    n=len(s)
    for i in range(n):
        for x in range(i+1,n+1):
            if len(s[i:x])==len(set(s[i:x])):
                mysubstr.append(s[i:x])
                mysubstr.sort(key=len)
                if mysubstr[-1]==s[i:x]:
                    startindex=i
                    endindex=x
    print(mystr[startindex:endindex])

subString(mystr.lower())

# METHOD 2: To check if a given string has repeated chars

#def hasRepeatedChars(s):
#    for i in range(len(s)):
#        if i != s.rfind(s[i]):
#            return True
#    return False

#def subString(s):
#    n=len(s)
#    for i in range(n):
#        for x in range(i+1,n+1):
#            temp=''.join(a for a in s[i:x])
#            if not hasRepeatedChars(temp):
```

```
#             mysubstr.append(s[i:x])
#             mysubstr.sort(key=len)
#             if mysubstr[-1]==s[i:x]:
#                 startindex=i
#                 endindex=x
#             print(mystr[startindex:endindex])
```

```
abhiSHEK
iSHEK
```

```
In [ ]:
```