Algorithms

December 21, 2018

1 Sliding Widow Technique

1.1 Count distinct elements in every window of size k

Tag: Sliding Window Technique, Hashtable. See [1].

```
Input:  arr[] = {1, 2, 1, 3, 4, 2, 3}, k = 4
Output: [3, 4, 4, 3]
```

We use the sliding window to update a hashtable, which maintains the distinct elements. And the time complexity is $O(n)$.

```
class Solution():
    '''2018-12-21
    '''
    def distinct(self,nums,k):
        if(not nums or len(nums)<k):
            return []
        d=dict()
        res=[]
        #init the first window
        for j in range(0,k):
            if(nums[j] not in d):
                d[nums[j]]=1
            else:
                d[nums[j]]+=1
        res.append(len(d))
```

[1] https://www.geeksforgeeks.org/count-distinct-elements-in-every-window-of-size-k/

```
        #update the remaining windows
        for i in range(1,len(nums)-k+1):
            #remove the first in the window, and add the last
            #to the window.
            first=nums[i-1]
            last=nums[i+k-1]
            if(d[first]==1):
                d.pop(first)
            else:
                d[first]-=1

            if(last not in d):
                d[last]=1
            else:
                d[last]+=1

            res.append(len(d))

        return res

    def testAll(self):
        testcase1={"nums":[1, 2, 1, 3, 4, 2, 3],"k":4,"expected":[3,4,4,3]}
        testcase2={"nums":[1, 2, 1],"k":4,"expected":[]}
        testcase3={"nums":[1, 2, 1, 3, 4, 2, 3, 5],"k":4,"expected":[3,4,4,3,4]}
        testcases=[testcase1,testcase2,testcase3]
        for testcase in testcases:
            self.test(testcase["nums"],testcase["k"],testcase["expected"])


    def test(self,nums,k,expected):
        res=self.distinct(nums,k)
        print("Test on nums={0}, k={1}. And {2} is expected, and {3} is got."\
                .format(nums,k,expected,res))


a=Solution()
a.testAll()
```

1.2 Sliding Window Maximum (Maximum of all subarrays of size k)

```
class Solution():
    """2018-12-21
    """
    def maxSlidingWindow(self,nums,k):
        pass
```