Sliding Window- variable size window

1. https://leetcode.com/problems/minimum-size-subarray-sum/description/?envType=problem-list-v2&envId=sliding-window

```
class Solution:
    def minSubArrayLen(self, target: int, nums: List[int]) ->
int:
        if(sum(nums))<target:
            return 0
        si=s=0
        m=len(nums)
        for li in range(len(nums)):
            s=s+nums[li]
            while s>=target:
                s=s-nums[si]
                m=min(m,li-si+1)
                si+=1
        return m
```

2. https://leetcode.com/problems/longest-substring-without-repeating-characters/description/?envType=problem-list-v2&envId=sliding-window

```
class Solution:
    def lengthOfLongestSubstring(self, s: str) -> int:
        si=0
        m=0
        vis=set()
        for li in range(len(s)):
            while s[li] in vis:
                vis.remove(s[si])
                si+=1
            vis.add(s[li])
            m=max(m,li-si+1)
        return m
```

3. https://leetcode.com/problems/maximize-the-confusion-of-an-exam/description/?envType=problem-list-v2&envId=sliding-window

```
class Solution:
    def maxConsecutiveAnswers(self, s: str, k: int) -> int:
        si=0
        m=0
        countT=0
```

```
        countF=0
        for li in range(len(s)):
            if s[li]=='T':
                countT+=1
            elif s[li]=='F':
                countF+=1
            while min(countT,countF)>k:
                if s[si]=='T':
                    countT-=1
                else:
                    countF-=1
                si+=1
            m=max(m,li-si+1)
        return m
```

4. timer

```
import turtle
import time

hr=int(turtle.numinput("timer","Hour"))
min=int(turtle.numinput("timer","minute"))
sec=int(turtle.numinput("timer","second"))

s=turtle.Turtle()
t=turtle.Turtle()
s.speed(0)
turtle.bgcolor("black")
s.color("white")
s.hideturtle()
s.pensize(2)
s.penup()
s.goto(-200,-50)
s.pendown()
s.forward(400)
s.left(90)
s.forward(100)
s.left(90)
s.forward(400)
s.left(90)
s.forward(100)

t.color("white")
t.hideturtle()
t.speed(0)
t.penup()
t.goto(-115,-35)
```

```
t.pendown()


while(1):

t.write(str(hr).zfill(2)+":"+str(min).zfill(2)+":"+str(sec).z
fill(2),font=("Arial", 60, "normal"))
    time.sleep(1)
    sec-=1
    if sec==-1:
        sec=59
        min-=1
    if min==-1:
        min=59
        hr-=1
    if hr==0 and min==0 and sec==0:
        break
    t.undo()

t.penup()
t.goto(-180,100)
t.pendown()
t.color("red")
t.write("Alert! Timer up",font=(("Arial", 40, "normal")))
turtle.mainloop()
```