Leetcode May Challenge DAY: 19

1. **Python**

```python
class StockSpanner(object):

    def __init__(self):
        self.st=[]
        self.d=0


    def next(self, price):
        """
        :type price: int
        :rtype: int
        """
        l=-1
        while self.st and self.st[-1][0]<=price:
            self.st.pop()
        if self.st:
            l=self.st[-1][-1]
        ans=self.d-l
        self.st.append((price,self.d))
        self.d+=1
        return ans
```

2. C++

```cpp
class StockSpanner {
public:
    stack<int> prices;
    stack<int> index;
    int span;
    StockSpanner() {
        span=0;
    }

    int next(int price) {
        span=1;
        while(prices.size()!=0 && prices.top()<=price){
            span+=index.top();
            prices.pop();
            index.pop();

        }
        prices.push(price);
        index.push(span);
        return span;

    }
};
```

3. JAVA

```java
class StockSpanner {

    Stack<int[]> st = new Stack<>();

    int idx = 0;

    public StockSpanner() {

        st.push(new int[]{-1, Integer.MAX_VALUE});

    }


    public int next(int price) {

        while (st.peek()[1] <= price) st.pop();

        int[] top = st.peek();

        st.push(new int[]{idx, price});

        return idx++ - top[0];

    }

}
```