```
P = [3,1,4,1,5,9,2,6,5,3,5,8,9,8]

class Stack:
      def __init__(self):
            self.items = []

      def push(self, item):
            self.items.append(item)

      def pop(self):
            return self.items.pop()

      def peek(self):

            return self.items[len(self.items)-1]

      def isEmpty(self):
            return self.items == []

      def size(self):
            return len(self.items)

def computeSpans1(P):
      S = [0] * len(P)                    → 2
      for i in range(0,len(P)):           → 2 + n(b + 3)
            k = 0                         → 1
            done = True                   → 1
            while ((k <=i) and done):     → 3   >
                  if (P[i-k] <= P[i]):    → 4   >  all run i^2 times
                        k = k+1           → 1   >
                  else:                   → never run under worst case
                        done = False      → never run under worst case
            S[i] = k                      → 1
      return S
                                          → 4 + 6n + 8 * sum(i = 1:n:i^2)
def computeSpans2(P):
      S = [0] * len(P)                    → 2
      D = Stack()                         → 1
      for i in range(0,len(P)):           → 2 + n(b + 3)
            h = 0                         → 1
            done = False                  → 1
            while ((D.isEmpty()==False) and (done==False)):      → 10 (5 for this
                                                                   line, twice)
                  if P[i] >= P[D.peek()]: → 4
                        D.pop()           → 1
                  else:                   → never run under worst case
                        done = True       → never run under worst case
            if D.isEmpty():               → 1
                  h = -1                  → 1
            else:                         → never run under worst case
                  h = D.peek()            → never run under worst case
            S[i] = (i-h)                  → 2
            D.push(i)                     → 1
      return S
                                          → 25n+5
print(computeSpans1(P))
print(computeSpans2(P))
                        → Wasn’t really sure how to deal with the while loops
                        → My paper submission is much more clear on my reasoning
```