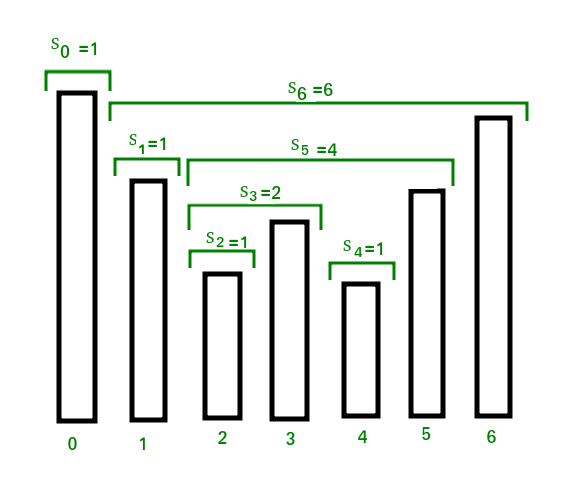
[The stock span problem](http://en.wikipedia.org/wiki/Stack_(abstract_data_type)" \l "The_Stock_Span_Problem" \t "_blank) is a financial problem where we have a series of n daily price quotes for a stock and we need to calculate span of stock’s price for all n days.   
The span Si of the stock’s price on a given day i is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the current day is less than its price on the given day.   
For example, if an array of 7 days prices is given as {100, 80, 60, 70, 60, 75, 85}, then the span values for corresponding 7 days are {1, 1, 1, 2, 1, 4, 6}



Solution:

static void calculateSpan(int price[], int n, int S[])

    {

        // Create a stack and push index of first element

        // to it

        Stack<Integer> st = new Stack<>();

        st.push(0);

        // Span value of first element is always 1

        S[0] = 1;

        // Calculate span values for rest of the elements

        for (int i = 1; i < n; i++) {

            // Pop elements from stack while stack is not

            // empty and top of stack is smaller than

            // price[i]

            while (!st.empty() && price[st.peek()] <= price[i])

                st.pop();

            // If stack becomes empty, then price[i] is

            // greater than all elements on left of it, i.e.,

            // price[0], price[1], ..price[i-1]. Else price[i]

            // is greater than elements after top of stack

            S[i] = (st.empty()) ? (i + 1) : (i - st.peek());

            // Push this element to stack

            st.push(i);

        }

    }