

# GeeksforGeeks

A computer science portal for geeks

[Android App](#) [GeeksQuiz](#)

## [Login/Register](#)

- [Home](#)
- [Algorithms](#)
- [DS](#)
- [GATE](#)
- [Interview Corner](#)
- [Q&A](#)
- [C](#)
- [C++](#)
- [Java](#)
- [Books](#)
- [Contribute](#)
- [Ask a Q](#)
- [About](#)

[Array](#)

[Bit Magic](#)

[C/C++](#)

[Articles](#)

[GFacts](#)

[Linked List](#)

[MCQ](#)

[Misc](#)

[Output](#)

[String](#)

[Tree](#)

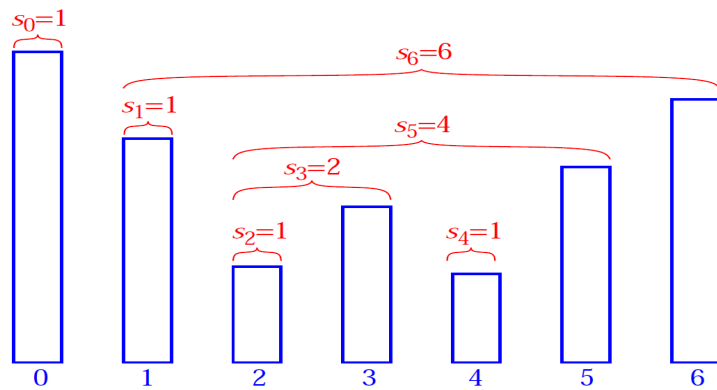
[Graph](#)

## The Stock Span Problem

[The stock span problem](#) 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  $S_i$  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 or equal to 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}



### A Simple but inefficient method

Traverse the input price array. For every element being visited, traverse elements on left of it and increment the span value of it while elements on the left side are smaller.

Following is implementation of this method.

// A brute force method to calculate stock span values

```
#include <stdio.h>
```

// Fills array S[] with span values

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

```
{
```

```
    // Span value of first day is always 1
```

```
    S[0] = 1;
```

```
    // Calculate span value of remaining days by linearly checking previous da
```

```
    for (int i = 1; i < n; i++)
```

```
    {
```

```
        S[i] = 1; // Initialize span value
```

```
        // Traverse left while the next element on left is smaller than price[i
```

```
        for (int j = i-1; (j>=0)&&(price[i]>=price[j]); j--)
```

```
            S[i]++;
```

```
    }
```

```
}
```

// A utility function to print elements of array

```
void printArray(int arr[], int n)
```

```
{
```

```
    for (int i = 0; i < n; i++)
```

```
        printf("%d ", arr[i]);
```

```
}
```

// Driver program to test above function

```
int main()
```

```
{
```

```
    int price[] = {10, 4, 5, 90, 120, 80};
```

```
    int n = sizeof(price)/sizeof(price[0]);
```

```
    int S[n];
```

```
    // Fill the span values in array S[]
```

```

    calculateSpan(price, n, S);

    // print the calculated span values
    printArray(S, n);

    return 0;
}

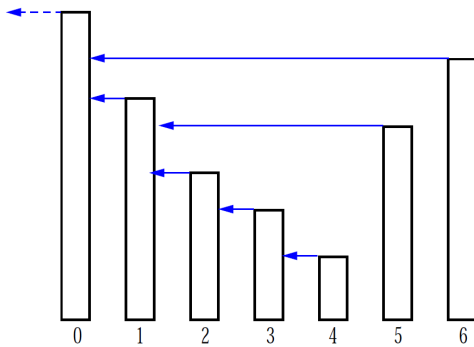
```

Time Complexity of the above method is  $O(n^2)$ . We can calculate stock span values in  $O(n)$  time.

### A Linear Time Complexity Method

We see that  $S[i]$  on day  $i$  can be easily computed if we know the closest day preceding  $i$ , such that the price is greater than on that day than the price on day  $i$ . If such a day exists, let's call it  $h(i)$ , otherwise, we define  $h(i) = -1$ .

The span is now computed as  $S[i] = i - h(i)$ . See the following diagram.



To implement this logic, we use a stack as an abstract data type to store the days  $i$ ,  $h(i)$ ,  $h(h(i))$  and so on. When we go from day  $i-1$  to  $i$ , we pop the days when the price of the stock was less than or equal to  $price[i]$  and then push the value of day  $i$  back into the stack.

Following is C++ implementation of this method.

```

// a linear time solution for stock span problem
#include <iostream>
#include <stack>
using namespace std;

// A brute force method to calculate stock span values
void calculateSpan(int price[], int n, int S[])
{
    // Create a stack and push index of first element to it
    stack<int> st;
    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.top()] < 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.top());

// Push this element to stack
st.push(i);
}
}

// A utility function to print elements of array
void printArray(int arr[], int n)
{
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
}

// Driver program to test above function
int main()
{
    int price[] = {10, 4, 5, 90, 120, 80};
    int n = sizeof(price)/sizeof(price[0]);
    int S[n];

    // Fill the span values in array S[]
    calculateSpan(price, n, S);

    // print the calculated span values
    printArray(S, n);

    return 0;
}

```

Output:

1 1 2 4 5 1

**Time Complexity:**  $O(n)$ . It seems more than  $O(n)$  at first look. If we take a closer look, we can observe that every element of array is added and removed from stack at most once. So there are total  $2n$  operations at most. Assuming that a stack operation takes  $O(1)$  time, we can say that the time complexity is  $O(n)$ .

**Auxiliary Space:**  $O(n)$  in worst case when all elements are sorted in decreasing order.

#### References:

[http://en.wikipedia.org/wiki/Stack\\_\(abstract\\_data\\_type\)#The\\_Stock\\_Span\\_Problem](http://en.wikipedia.org/wiki/Stack_(abstract_data_type)#The_Stock_Span_Problem)  
<http://crypto.cs.mcgill.ca/~crepeau/CS250/2004/Stack-I.pdf>

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

## Related Topics:

- [Visa Interview Experience | Set 6 \(On-Campus\)](#)
- [Count number of ways to reach a given score in a game](#)
- [Greedy Algorithm for Egyptian Fraction](#)
- [Find length of period in decimal value of 1/n](#)
- [How to check if an instance of 8 puzzle is solvable?](#)
- [How to Implement Forward DNS Look Up Cache?](#)
- [Birthday Paradox](#)
- [Job Sequencing Problem | Set 1 \(Greedy Algorithm\)](#)

Tags: [stack](#)

Like 16

Tweet 4

 1

Writing code in comment? Please use [ideone.com](#) and share the link here.

36 Comments

GeeksforGeeks

 Login ▾

 Recommend  Share

Sort by Newest ▾



Join the discussion...



**PORUS** • a month ago

How about below code? No Extra Space!!

```
#include <stdio.h>
```

```
int main(void) {
```

```
// your code goes here
```

```
int arr[7] = {100, 65, 60, 70, 60, 75, 101};
```

```
int S[7];
```

```
int i,j,count;
```

```
int n = 7;
```

```
S[0] = 1;
```

```
printf("%d ",S[0]);
```

```
for(i=1;i<n;i++){ if(arr[i]<=arr[i-1]) s[i]=1; else { count=1; j=i-1; while(j>=0
```

```
&& arr[j] <= arr[i]){
```

```
count = count + S[j];
```

```
j = j - S[j];
```

```
}
```

```
S[i] = count;
```

```

    S[i] = count;
}
printf("%d ",S[i]);
}
printf("\n");
return 0;
}

```

^ | v • Reply • Share ›



**Ashish Jaiswal** • 2 months ago

Above algo is not working in case `int arr[]={1,3,2};`  
 ans: 1,2,1 but it gives outout 1,2,2..will any one explain please [@neelabhsingh](#) [@GOPI GOPINATH](#) please....

^ | v • Reply • Share ›



**Hitesh Saini** → Ashish Jaiswal • a month ago

it is giving 1,2,1 as output  
 check again.!

<https://ideone.com/xxVV2J>

^ | v • Reply • Share ›



**Joey** • 2 months ago

Will this algo work in all cases ? I know it is not space efficient as it uses another extra stack.

My Algo is :

i.Create two stacks and one int array span (all initialised to 1).

ii.While pushing next element check if the price at the stack top of 1st stack is greater or smaller than the next element.

i.if price of next element is smaller than stack 1 top,push the element into Stack 1.

ii.else if the price of next element is greater than stack 1 top,then pop elements from stack 1 and push into stack 2

till elements in stack 1 is smaller than the next element.While popping,increment the span of that element.Then pop all elements from stack 2 and push back into stack1.

Continue the while loop till all inputs are scanned.

^ | v • Reply • Share ›



**shubham** • 2 months ago

Please correct the conditions in the while loop in the second approach! it should be :

`price[sk.top()] <= price[i]`

^ | v • Reply • Share ›

**rajdeep** • 3 months ago

while (!st.empty() && price[st.top()] <= price[i])  
 st.pop();  
 should check for <= not only <  
 since price can be same on consecutive days !!  
 1 ^ | v • Reply • Share ›

**Young Han Lee** • 3 months ago

I think this can be solved not using stack in  $O(n)$  time complexity and  $O(1)$  auxiliary space.

<http://ideone.com/Xcqjje>

```
void calculateSpan(int price[], int n, int S[]) {
  S[0] = 1;

  for (int i = 1; i < n; i++) {
    int j = i - 1;

    while (j >= 0 && price[j] <= price[i]) j -= S[j];

    S[i] = i - j;
  }
}
```

^ | v • Reply • Share ›

**Guest** ➔ Young Han Lee • 2 months ago

This is brute force, and it will take  $O(n^2)$  time when prices are arranged in sorted order.  
 ^ | v • Reply • Share ›

**PORUS** ➔ Guest • a month ago

No Even this will take  $O(2n)$  time in worst case..  
 Worst case is 5 4 3 2 1 6  
 ^ | v • Reply • Share ›

**Guest** • 4 months ago

Can any one help how is it  $O(n)$  when the sequence is 10,8,6,4,11. while loop runs almost  $n$  times at the end and I think it is  $O(n^2)$ .  
 ^ | v • Reply • Share ›

**Guest** • 4 months ago

10,8,6,4,3,11. In this case the while loop runs for almost  $n$  times. How is the time complexity  $O(n)$ . I think it is  $O(n^2)$ . Please help  
 ^ | v • Reply • Share ›

**Amogh Margoor** • 4 months ago

This can be mapped to problem of finding next greater element in an array  
(<http://www.geeksforgeeks.org/n...>)

Solution is to find for each element 'i', the next greater element 'j' in reverse order i.e.,  $j < i$  and no element between j and i will be greater than element at i.

Span for element at 'i' is  $j-i+1$  as all the elements between j and i are smaller than element at 'i'.

Following is the Code:

<http://ideone.com/CsU8cZ>

^ | v • Reply • Share ›

**Ramshah** • 4 months ago

This Animation would be useful to understand Above Problem

<http://csanimated.wordpress.co...>

2 ^ | v • Reply • Share ›

**helper** • 4 months ago

my java code <http://ideone.com/uc6kfL>

^ | v • Reply • Share ›

**K. M. Fazle Azim** • 6 months ago

The definition of the problem seems to be incorrect. It should be

"The span  $S_i$  of the stock's price on a given day i is defined as the maximum number of consecutive days up to (i.e. including) the given day, for which the price of the stock on each of those days is less than or equal to its price on the given day i."

Also the wiki link does not have any reference for this problem.

1 ^ | v • Reply • Share ›

**Coder M** • 7 months ago

<http://ideone.com/68STEK>

1 ^ | v • Reply • Share ›

**Guest** • 7 months ago

In stack based method, the while condition should be:

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

otherwise for 60,70,70,70

output will be: 1 2 1 1

^ | v • Reply • Share ›

**Guest** ➔ Guest • 7 months ago





Also st.push(0) should be commented

^ | v • Reply • Share ›



**pradeep kumar** • 8 months ago

How does it works??What its definition??

```
stack<int> st;
```

^ | v • Reply • Share ›



**Amit Jambotkar** • 10 months ago

```
public class Stack<t extends="" number=""> {
```

```
int top;
```

```
int capacity;
```

```
T values[];
```

```
@SuppressWarnings("unchecked")
```

```
public Stack(int capacity){
```

```
this.values= (T[])new Number[capacity] ;
```

```
this.capacity=capacity;
```

```
this.top=-1;
```

```
}
```

```
void push(T value){
```

[see more](#)

^ | v • Reply • Share ›



**sijayaraman** • a year ago

```
void stock(int arr[],int n)
```

```
{
```

```
int index=0;
```

```
int result[n];
```

```
while(index<n) {="" result[index]="1;" for(int="" j="index;j">=0;j--)
```

```
{
```

```
if(arr[index]>arr[j])
```

```
{
```

```
result[index]++;
```

```
}
```

```

,
else if(arr[j]> arr[index])
{
break;
}
}
index++;
}

```

//output

```

for(int i=0;i<n;i++) {="" cout<<result[i]<<" ";="" }="" int="" main()="" {="" int="" arr[]={10,"
4,="" 5,="" 90,="" 120,="" 80};="" int="" n="" sizeof(arr)/sizeof(arr[0]);="" stock(arr,n);="" }="" >

```

^ | v • Reply • Share ›



**Guest** • a year ago

Does this work for {2,3,10,4,5,6,7}?

^ | v • Reply • Share ›



**sapy** • 2 years ago

I think the code needs correction...

```

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

```

4 ^ | v • Reply • Share ›



**Sairam Ravu** • 2 years ago

```
int price[] = {10, 4, 5, 90, 120, 80};
```

For this input the span[5] that is for 80 should be 3 but your answer is 1.

I didn't understand the brute force method, it just checks for less than or equal it doesn't check for consecutive, if so then for {10,4,5,90,120,80}, span[5] will be 4 instead of 3 because you are just incrementing the count for how many values which are less than given value.

```

/* Paste your code here (You may delete these lines if not writing code) */

```

^ | v • Reply • Share ›



**Sampath Kumar** → Sairam Ravu • 5 months ago

You got it wrong Ravu

^ | v • Reply • Share ›



**GeeksforGeeks** → Sairam Ravu • 2 years ago

Please take a closer look at the problem statement, especially following part.

"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

consecutive days ~~just before the given day~~, for which the price of the stock on the current day is less than or equal to its price on the given day."

^ | v • Reply • Share ›



**AlexC** • 2 years ago

this is the same algorithm (with a small change) used to efficiently construct Cartesian tree.

<http://en.wikipedia.org/wiki/A...>

<http://en.wikipedia.org/wiki/C...>

^ | v • Reply • Share ›



**nishant** • 2 years ago

I think this works.

```
void calculateSpan(int price[], int n, int S[])
{
    int i,j;
    S[0]=1;
    for(i=1;i<n;i++)
    {
        j=i-1;
        while(price[i]>price[j])
            j=j-S[j];
        S[i]=i-j;
    }
}
```

^ | v • Reply • Share ›



**GeeksforGeeks** • 2 years ago

This doesn't seem to work for given example {100, 80, 60, 70, 60, 75, 85}. The span value of index 4 is 1, but span value of index 5 is 4, not 2.

^ | v • Reply • Share ›



**GeeksforGeeks** • 2 years ago

This doesn't seem to work for given example {100, 80, 60, 70, 60, 75, 85}. The span value of index 4 is 1, but span value of index 5 is 4, not 2.

^ | v • Reply • Share ›



**Anand Kumar** • 2 years ago

// Fills array S[] with span values.

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

{

// Span value of first day is always 1.

```
S[0] = 1;.
```

```
// Calculate span value of remaining days by using the span values of previous days.
```

```
for (int i = 1; i < n; i++).
```

```
{.
```

```
if(price[i]>=price[i-1]) S[i]=S[i-1]+1;.
```

```
else S[i]=1;.
```

```
}.  
}
```

^ | v • Reply • Share ›



**Shashank Rai** → Anand Kumar • a year ago

It is wrong, v don't hav 2 print length of non-decreasing prices upto the given day. eg. 10 4 5 90, den fr 90 ur algo gives 3 but d ans shd b 4 since 10 < 90 even though 10 > 4

^ | v • Reply • Share ›



**Anand Kumar** • 2 years ago

```
// Fills array S[] with span values.
```

```
void calculateSpan(int price[], int n, int S[]).
```

```
{
```

```
// Span value of first day is always 1.
```

```
S[0] = 1;.
```

```
// Calculate span value of remaining days by using the span values of previous days.
```

```
for (int i = 1; i < n; i++).
```

```
{.
```

```
if(price[i]>=price[i-1]) S[i]=S[i-1]+1;.
```

```
else S[i]=1;.
```

```
}.  
}
```

^ | v • Reply • Share ›



**kx** → Anand Kumar • a year ago

It looks good... but this won't work..

After : if(price[i]>=price[i-1]) S[i]=S[i-1]+1;

you need to additionally check for a similar condition: (price[i]>=price[i-S[i]]), and update S[i] again accordingly based on the result => and continue this step until you reach the top of the array. (In a way its similar to the stack implementation above).

Else, your case will fail for something like: price = {70,60,80}. S[2] here should be 3; your method will give 2 (because you stop after comparing with 60 and do not check

your method will give 2 (because you stop after comparing with 60, and do not check against 70].

^ | v • Reply • Share ›



**panda** • 2 years ago

lol, i was interviewed by storm8 with this question....

1 ^ | v • Reply • Share ›



**shashank** → panda • 2 years ago

Hi panda , what the question in coding challenge ?

Thanks

```
/* Paste your code here (You may delete these lines if not writing code) */
```

^ | v • Reply • Share ›

 [Subscribe](#)

 [Add Disqus to your site](#)

 [Privacy](#)



GeeksforGeeks

Like

92,235 people like GeeksforGeeks.



Feedback social media

- 
- 
- - [Interview Experiences](#)
  - [Advanced Data Structures](#)
  - [Dynamic Programming](#)
  - [Greedy Algorithms](#)
  - [Backtracking](#)
  - [Pattern Searching](#)
  - [Divide & Conquer](#)
  - [Mathematical Algorithms](#)
  - [Recursion](#)
  - [Geometric Algorithms](#)
- 

## • Popular Posts

- [All permutations of a given string](#)
- [Memory Layout of C Programs](#)
- [Understanding “extern” keyword in C](#)
- [Median of two sorted arrays](#)
- [Tree traversal without recursion and without stack!](#)
- [Structure Member Alignment, Padding and Data Packing](#)
- [Intersection point of two Linked Lists](#)
- [Lowest Common Ancestor in a BST](#)
- [Check if a binary tree is BST or not](#)
- [Sorted Linked List to Balanced BST](#)

Follow @GeeksforGeeks

[Subscribe](#)

## • Recent Comments

- [Humble\\_Learner](#)

The question itself is about without the use of...

[Linked complete binary tree & its creation](#) · [34 minutes ago](#)

- Haikent

```
bool isbst(node<t> *rm ,int min=INT_MIN,int...
```

[A program to check if a binary tree is BST or not](#) · [41 minutes ago](#)

- Haikent

int isbst(struct node \*tm) {...

[A program to check if a binary tree is BST or not](#) · [1 hour ago](#)

- [Gaurav pruthi](#)

yup..... Author is trying to explain...

[When are static objects destroyed?](#) · [3 hours ago](#)

- [Nitin Pandey](#)

Round 1 - Question 2

<https://ideone.com/5A8Ffv>

[LinkedIn Interview Experience | Set 5 \(On campus\)](#) · [3 hours ago](#)

- [Vito](#)

To print the sub-array also, please note this...

[Largest Sum Contiguous Subarray](#) · [3 hours ago](#)

•

@geeksforgeeks, [Some rights reserved](#) [Contact Us!](#)

Powered by [WordPress](#) & [MooTools](#), customized by geeksforgeeks team