Next Greater Element

Given an array, print the Next Greater Element (NGE) for every element. The Next greater Element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1.

Examples:
a) For any array, rightmost element always has next greater element as -1.
b) For an array which is sorted in decreasing order, all elements have next greater element as -1.
c) For the input array [4, 5, 2, 25}, the next greater elements for each element are as follows.

```
Element       NGE
   4      -->   5
   5      -->   25
   2      -->   25
   25     -->   -1
```

d) For the input array [13, 7, 6, 12}, the next greater elements for each element are as follows.

```
  Element        NGE
   13      -->    -1
   7       -->     12
   6       -->     12
   12      -->     -1
```

Method 1 (Simple)
Use two loops: The outer loop picks all the elements one by one. The inner loop looks for the first greater element for the element picked by outer loop. If a greater element is found then that element is printed as next, otherwise -1 is printed.

Thanks to Sachin for providing following code.

```
pchild for suggesting following approach.<br>
1) Push the first element to stack.<br>
2) Pick rest of the elements one by one and follow following steps in
....a) Mark the current element as <em>next</em>.<br>
....b) If stack is not empty, then pop an element from stack and compa
....c) If next is greater than the popped element, then <em>next </em>
....d) Keep poppoing from the stack while the popped element is smalle
....g) If <em>next </em>is smaller than the popped element, then push
3) After the loop in step 2 is over, pop all the elements from stack a
<pre class="brush: cpp; highlight: [52,53,54,55,56,57,58,59,60,61,62,6
#include<stdlib.h>
#define STACKSIZE 100

// stack structure
struct stack
{
    int top;
    int items[STACKSIZE];
};

// Stack Functions to be used by printNGE()
void push(struct stack *ps, int x)
{
    if (ps->top == STACKSIZE-1)
    {
        printf("Error: stack overflow\n");
        getchar();
        exit(0);
    }
    else
    {
        ps->top += 1;
        int top = ps->top;
        ps->items [top] = x;
    }
}

bool isEmpty(struct stack *ps)
{
    return (ps->top == -1)? true : false;
}

int pop(struct stack *ps)
{
```


Popular Posts

```
    int temp;
    if (ps->top == -1)
    {
        printf("Error: stack underflow \n");
        getchar();
        exit(0);
    }
    else
    {
        int top = ps->top;
        temp = ps->items [top];
        ps->top -= 1;
        return temp;
    }
}

/* prints element and NGE pair for all elements of
arr[] of size n */
void printNGE(int arr[], int n)
{
    int i = 0;
    struct stack s;
    s.top = -1;
    int element, next;

    /* push the first element to stack */
    push(&s, arr[0]);

    // iterate for rest of the elements
    for (i=1; i<n; i++)
    {
        next = arr[i];

        if (isEmpty(&s) == false)
        {
            // if stack is not empty, then pop an element from stack
            element = pop(&s);

            /* If the popped element is smaller than next, then
                a) print the pair
                b) keep popping while elements are smaller and
                stack is not empty */
            while (element < next)
            {
                printf("\n %d --> %d", element, next);
                if(isEmpty(&s) == true)
                   break;
```

```
                element = pop(&s);
            }

            /* If element is greater than next, then push
               the element back */
            if (element > next)
                push(&s, element);
        }

        /* push next to stack so that we can find
           next greater for it */
        push(&s, next);
    }

    /* After iterating over the loop, the remaining
       elements in stack do not have the next greater
       element, so print -1 for them */
    while (isEmpty(&s) == false)
    {
        element = pop(&s);
        next = -1;
        printf("\n %d --> %d", element, next);
    }
}

/* Driver program to test above functions */
int main()
{
    int arr[]= {11, 13, 21, 3};
    int n = sizeof(arr)/sizeof(arr[0]);
    printNGE(arr, n);
    getchar();
    return 0;
}
</pre>
<p>Output:</p>
<pre> 11 --> 13
 13 --> 21
 3 --> -1
 21 --> -1
</pre>
<p>Time Complexity: O(n). The worst case occurs when all elements are
a) Initialy pushed to the stack.<br>
b) Popped from the stack when next element is being processed.<br>
c) Pushed back to the stack because next element is smaller.<br>
d) Popped from the stack in step 3 of algo.</p>
<p>Source:<br>
```


Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 26 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 30 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 55 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 57 minutes ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

```
<a href="http://geeksforgeeks.org/forum/topic/next-greater-element#pos
<p>Please write comments if you find the above codes/algorithms incorr
```

```
<script async="" src="//pagead2.googlesyndication.com/pagead/js/adsbyg
<!-- Big Rectangle Blog Bottom -->
<ins class="adsbygoogle" style="display:inline-block;width:336px;heigh
<script>
(adsbygoogle = window.adsbygoogle || []).push({});
</script>
```



```
<h3>Related Tpoics:</h3><ul><li><a href="http://www.geeksforgeeks.org/
<iframe src="http://www.facebook.com/plugins/like.php?href=http%3A%2F%
<iframe id="twitter-widget-0" scrolling="no" frameborder="0" allowtran
<div style="text-indent: 0px; margin-top: 0px; margin-right: 0px; marg
<!-- Place this render call where appropriate -->
<script type="text/javascript">
  (function() {
    var po = document.createElement('script'); po.type = 'text/javascr
    po.src = 'https://apis.google.com/js/plusone.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.i
  })();
</script></div>
```

Writing code in comment? Please use **ideone.com** and share the link here.