Replace every element with the next greatest

Given an array of integers, replace every element with the next greatest element (greatest element on the right side) in the array. Since there is no element next to the last element, replace it with -1. For example, if the array is {16, 17, 4, 3, 5, 2}, then it should be modified to {17, 5, 5, 5, 2, -1}.

The question is very similar to this post and solutions are also similar.

A **naive method** is to run two loops. The outer loop will one by one pick array elements from left to right. The inner loop will find the greatest element present after the picked element. Finally the outer loop will replace the picked element with the greatest element found by inner loop. The time complexity of this method will be O(n*n).
A **tricky method** is to replace all elements using one traversal of the array. The idea is to start from the rightmost element, move to the left side one by one, and keep track of the maximum element. Replace every element with the maximum element.

```
#include <stdio.h>

/* Function to replace every element with the
   next greatest element */
void nextGreatest(int arr[], int size)
{
  // Initialize the next greatest element
  int max_from_right =  arr[size-1];

  // The next greatest element for the rightmost element
  // is always -1
  arr[size-1] = -1;

  // Replace all other elements with the next greatest
  for(int i = size-2; i >= 0; i--)
```


```
  {
    // Store the current element (needed later for updating
    // the next greatest element)
    int temp = arr[i];

    // Replace current element with the next greatest
    arr[i] = max_from_right;

    // Update the greatest element, if needed
    if(max_from_right < temp)
       max_from_right = temp;
  }
}

/* A utility Function that prints an array */
void printArray(int arr[], int size)
{
  int i;
  for (i=0; i < size; i++)
    printf("%d ", arr[i]);
  printf("\n");
}

/* Driver program to test above function */
int main()
{
  int arr[] = {16, 17, 4, 3, 5, 2};
  int size = sizeof(arr)/sizeof(arr[0]);
  nextGreatest (arr, size);
  printf ("The modified array is: \n");
  printArray (arr, size);
  return (0);
}
```

Output:

```
The modified array is:
17 5 5 5 2 -1
```

Time Complexity: O(n) where n is the number of elements in array.

Please write comments if you find any of the above codes/algorithms incorrect, or find other ways to solve the same problem.

Popular Posts

Related Tpoics:

- Remove minimum elements from either side such that 2*min becomes more than max
- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
- Find the number of zeroes
- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3

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

21 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

sudhanshu • 3 months ago

http://stackoverflow.com/quest...

best approach.....:)

Reply • Share ›

sandeep • 6 months ago

Is output shown in test case correct?

{16, 17, 4, 3, 5, 2}, sould be {17,17,5,5,5,-2}

Reply • Share ›

Alok Kumar • 9 months ago

Oh..or may be I got the point....we have to start from right..find the next greates update the array...and then proceed to next element.....so as the source array solution is correct...However little more clarity on the solution wazs required.

Reply • Share ›

Alok Kumar • 9 months ago

Oh..or may be I got the point....we have to start from right..find the next greates update the array...and then proceed to next element.....so as the source array solution is correct...However little more clarity on the solution wazs required.

Reply • Share ›

Alok Kumar • 9 months ago

Above solution is wrong..try run it on the array {18,14,8,17,25}....the aser shou solution gives {25,25,25,25,-1}.....I think you are finding greatest number on rig greatER number..plz have a review.

1 Reply • Share ›

705 Subscribe

Recent Comments

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

Write a C program to Delete a Tree. · 6 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 10 minutes ago

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

Root to leaf path sum equal to a given number · 35 minutes ago

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

Count trailing zeroes in factorial of a number · 37 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

Saurabh Pandey ➦ Alok Kumar • 9 months ago

The solution for {18,14,8,17,25} is {25,25,25,25,-1} because last eleme will be replaced by 25.

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

  • Reply • Share ›

Alok Kumar • 9 months ago

Above solution is wrong..try run it on the array {18,14,8,17,25}....the aser shou solution gives {25,25,25,25,-1}.....I think you are finding greatest number on ri greatER number..plz have a review.

  • Reply • Share ›

kaushik • 10 months ago

```
//Kaushik Sahu

#include<stdio.h>
#define len sizeof(arr)/sizeof(arr[0])



void nextMax(int arr[],int size)
{
    int i,max,temp;
    max = arr[size-1];
    arr[size-1] = -1;
    temp = max;

    for(i=size-2;i>=0;i--)
    {
        if(arr[i] > max)
```

AdChoices

▶ Element Seven

▶ Array Replace

▶ Array Element

AdChoices

▶ Replace Java

▶ Replace

▶ Element To

```
            temp = arr[i];
```

see more

Reply • Share ›

Sanjay Garg • 11 months ago

next greatest element and greatest element on right side are two different thing

Reply • Share ›

abhishek08aug • a year ago

```
#include<stdio.h>

int replace_with_next_greatest_element(int array[], int n) {
  int i, next_greatest_element, current_element;
  for(i=n-1, next_greatest_element=-1; i>=0; i--) {
    current_element=array[i];
    array[i]=next_greatest_element;
    if(current_element>next_greatest_element) {
      next_greatest_element=current_element;
    }
  }
}

int main(){
  int array[]={16, 17, 4, 3, 5, 2};
  int n=sizeof(array)/sizeof(array[0]);
  int i;
  for(i=0; i<n; i++) {
```

see more

Reply • Share ›

Ganesh • a year ago

[sourcecode language="JAVA"]
public class ReplaceEveryElementWithTheNextGreatest {

public static void main(String[] args) {
int arr[] = {16, 17, 4, 3, 5, 2};
int max = -1;
for(int i = arr.length - 1; i >= 0; i--) {
int temp = arr[i];
arr[i] = max;
if (max < temp) max = temp;
}
for (int i : arr) {
System.out.print(i + " ");
}
}
}

• Reply • Share ›

Himanshu • 2 years ago

The code should not work for 1 2 3 4 5 6.
The output computed would be 6 6 6 6 6 -1.
Whereas the expected one is 2 3 4 5 6 -1.
Am I missing on something?

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

• Reply • Share ›

Kartik ➚ Himanshu • 2 years ago

It is next greatest, not next greater.

Saran • 2 years ago

```
#include
#include
void main()
{
clrscr();
int a[50],i,j,max,n;
cout<>n;
cout<<"Enter the elements for the array:";
for(i=1;i>a[i];
for(i=2;i<=n;i++)
{
max=a[i];
for(j=i+1;jmax)
max=a[j];
}
a[i-1]=max;
}
a[n]=-1;
for(i=1;i<=n;i++)
cout<<"\nThe elements are:"<<a[i];
getch();
}
```


Psycho • 2 years ago

```
#include <stdio.h>

void printList (int arr[], int size) {
```

```
  int i ;
  printf ( "\n" ) ;
  for ( i = 0 ; i < size ; i ++ )
    printf ( "%d ", arr[i] ) ;
  printf ( "\n" );
}

int main () {
  int arr[] = {4, 12, 43, 3, 2, 9, 4, 12, 2, 8, 0} ;
  int size = sizeof(arr) / sizeof(arr[0]) ;
  int i, max = arr[size-1] ;
  printList (arr, size) ;
  for ( i = size-2 ; i >= 0 ; i -- ) {
    if ( arr[i] <= max )
      arr[i] = max ;
    else max = arr[i] ;
  }
  arr[size-1] = -1 ;
  printList (arr, size) ;
  return 0 ;
}
```


kartik • 2 years ago

```
Your code has some problem..i solved it...have a look at if else cond
void nextGreatest(int arr[], int size)
{
  // Initialize the next greatest element
  int max_from_right =  arr[size-1];
```

```
// The next greatest element for the rightmost element
// is always -1
arr[size-1] = -1;

// Replace all other elements with the next greatest
for(int i = size-2; i >= 0; i--)
{
  // Store the current element (needed later for updating
  // the next greatest element)
  int temp = arr[i];
```

see more

• Reply • Share ›

GeeksforGeeks ➚ kartik • 2 years ago

@kartik: Could you please post an example input for which the original

• Reply • Share ›

Ram • 2 years ago

This solution is wrong.

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

• Reply • Share ›

Ram ➚ Ram • 2 years ago

it doesnot work for {1,3,7} .. it gives {7,7,-1} it must be {3,7,-1}

• Reply • Share ›

kartik ➚ Ram • 2 years ago

@Ram: Please take a closer look at the problem statement. It s
element.

• Reply • Share ›

chinna • 2 years ago

hi iam reddy.this is very usefull for me.

• Reply • Share ›

Subscribe Add Disqus to your site