

GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- [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](#)

Sort an array in wave form

Given an unsorted array of integers, sort the array into a wave like array. An array 'arr[0..n-1]' is sorted in wave form if $arr[0] \geq arr[1] \leq arr[2] \geq arr[3] \leq arr[4] \geq \dots$

Examples:

Input: arr[] = {10, 5, 6, 3, 2, 20, 100, 80}
Output: arr[] = {10, 5, 6, 2, 20, 3, 100, 80} OR
 {20, 5, 10, 2, 80, 6, 100, 3} OR
 any other array that is in wave form

Input: arr[] = {20, 10, 8, 6, 4, 2}
Output: arr[] = {20, 8, 10, 4, 6, 2} OR
 {10, 8, 20, 2, 6, 4} OR

any other array that is in wave form

Input: arr[] = {2, 4, 6, 8, 10, 20}

Output: arr[] = {4, 2, 8, 6, 20, 10} OR
any other array that is in wave form

Input: arr[] = {3, 6, 5, 10, 7, 20}

Output: arr[] = {6, 3, 10, 5, 20, 7} OR
any other array that is in wave form

We strongly recommend to minimize your browser and try this yourself first.

A **Simple Solution** is to use sorting. First sort the input array, then swap all adjacent elements.

For example, let the input array be {3, 6, 5, 10, 7, 20}. After sorting, we get {3, 5, 6, 7, 10, 20}. After swapping adjacent elements, we get {5, 3, 7, 6, 20, 10}. Below is C++ implementation of this simple approach.

```
// A C++ program to sort an array in wave form using a sorting function
```

```
#include<iostream>
```

```
#include<algorithm>
```

```
using namespace std;
```

```
// A utility method to swap two numbers.
```

```
void swap(int *x, int *y)
```

```
{
```

```
    int temp = *x;
```

```
    *x = *y;
```

```
    *y = temp;
```

```
}
```

```
// This function sorts arr[0..n-1] in wave form, i.e.,
```

```
// arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4] >= arr[5]..
```

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

```
{
```

```
    // Sort the input array
```

```
    sort(arr, arr+n);
```

```
    // Swap adjacent elements
```

```
    for (int i=0; i<n-1; i += 2)
```

```
        swap(&arr[i], &arr[i+1]);
```

```
}
```

```
// Driver program to test above function
```

```
int main()
```

```
{
```

```
    int arr[] = {10, 90, 49, 2, 1, 5, 23};
```

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

```
    sortInWave(arr, n);
```

```
    cout << "Sorted array \n";
```

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

```
        cout << arr[i] << " ";
```

```
    return 0;
```

```
}
```

Output:

Sorted array
2 1 10 5 49 23 90

The time complexity of the above solution is $O(n \log n)$ if a $O(n \log n)$ sorting algorithm like [Merge Sort](#), [Heap Sort](#), .. etc is used.

This can be done in **$O(n)$ time by doing a single traversal** of given array. The idea is based on the fact that if we make sure that all even positioned (at index 0, 2, 4, ..) elements are greater than their adjacent odd elements, we don't need to worry about odd positioned element. Following are simple steps.

1) Traverse all even positioned elements of input array, and do following.

....a) If current element is smaller than previous odd element, swap previous and current.

....b) If current element is smaller than next odd element, swap next and current.

Below is C++ implementation of above simple algorithm.

```
// A O(n) program to sort an input array in wave form
#include<iostream>
using namespace std;

// A utility method to swap two numbers.
void swap(int *x, int *y)
{
    int temp = *x;
    *x = *y;
    *y = temp;
}

// This function sorts arr[0..n-1] in wave form, i.e., arr[0] >=
// arr[1] <= arr[2] >= arr[3] <= arr[4] >= arr[5] ....
void sortInWave(int arr[], int n)
{
    // Traverse all even elements
    for (int i = 0; i < n; i+=2)
    {
        // If current even element is smaller than previous
        if (i>0 && arr[i-1] > arr[i] )
            swap(&arr[i], &arr[i-1]);

        // If current even element is smaller than next
        if (i<n-1 && arr[i] < arr[i+1] )
            swap(&arr[i], &arr[i + 1]);
    }
}

// Driver program to test above function
int main()
{
    int arr[] = {10, 90, 49, 2, 1, 5, 23};
    int n = sizeof(arr)/sizeof(arr[0]);
    sortInWave(arr, n);
    cout << "Sorted array \n";
```

```
for (int i=0; i<n; i++)  
    cout << arr[i] << " ";  
return 0;  
}
```

Output:

Sorted array
90 10 49 1 5 2 23

This article is contributed by **Shivam**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Related Topics:

- [Find Union and Intersection of two unsorted arrays](#)
- [Pythagorean Triplet in an array](#)
- [Maximum profit by buying and selling a share at most twice](#)
- [Design a data structure that supports insert, delete, search and getRandom in constant time](#)
- [Print missing elements that lie in range 0 – 99](#)
- [Iterative Merge Sort](#)
- [Group multiple occurrence of array elements ordered by first occurrence](#)
- [Given a sorted and rotated array, find if there is a pair with a given sum](#)



Tweet

6

g+1

2

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

16 Comments

GeeksforGeeks

1 Login ▾

♥ Recommend

🔗 Share

Sort by Newest ▾



Join the discussion...

sanju • 3 days ago

What about the duplicate elements or the array which can't be sorted in wave form at all.

^ | ▾ • Reply • Share ›



nick • 4 months ago

Good question.

^ | ▾ • Reply • Share ›

Evil Mad Scientist • 4 months ago

I don't think the proposed $O(n)$ solution is correct. It doesn't work for this input: {1, 1, 1, 1, 1, 2, 2, 2, 2}. The correct output would be {2, 1, 2, 1, 2, 1, 2, 1, 2, 1}, while the code above will return {1, 1, 1, 1, 2, 1, 2, 2, 2, 2}

^ | v • Reply • Share ›



dumind → Evil Mad Scientist • 4 months ago

The output which code will return is correct too... There can be many outputs that satisfy the constraint specified for the wave form array.

^ | v • Reply • Share ›

Siya → dumind • 4 months ago

Though there are many outputs which satisfy the constraints, the above example is not following the constraints, it is not making wave. Infact with repetition none of the above algorithm is working correctly.

we can do this by

- 1.sort the array($n \log n$)
- 2.Take another array with same space
- 3.populate even indexes first (with sorted array elements) and then odd time- ($n \log n$)

space- $O(n)$

3 ^ | v • Reply • Share ›

Aditya Goel → Siya • 3 months ago

We don't need an array with same space. We can do that even by taking half space.

^ | v • Reply • Share ›



puneet • 4 months ago

Using $O(n)$ median finding algo, first find median. Then in another array, put median at first position. Then put first elem of left part of median at 2nd position and 1st elem of right part at 3rd position. Similarly 2nd element of both part and so on

^ | v • Reply • Share ›

rverma • 4 months ago

this is $O(n)$ solution below

<http://ideone.com/di4E9P>

^ | v • Reply • Share ›



sivakiran • 4 months ago

<http://ideone.com/nw0MYF>

works both for odd and even input

^ | v • Reply • Share ›

rihansh • 4 months ago

here is O(N) simple implementation of the above problem handling all kinds of case ..

<http://ideone.com/Uqpuyz>

^ | v • Reply • Share ›

Ganesh Vijay ➔ rihansh • 4 months ago

I think your program needs some tweaking. For example, you are storing values from 1 to N. And in First iteration of Convert function you are comparing arr[1] with arr[0]. arr[0] has no scope. It might fail for test cases

^ | v • Reply • Share ›

rihansh ➔ Ganesh Vijay • 4 months ago

Hello dude ,

As i declared global array this value will be zero and i am assuming that all the values given will be strictly greater than zero ..

^ | v • Reply • Share ›

rihansh • 4 months ago

Here is one O(nlogn) simple implementation ..

<http://ideone.com/pdl3ew>

1 ^ | v • Reply • Share ›

Ganesh Vijay • 4 months ago

The outputs of the C++ implementation are incorrect! It should be changed based on the input in the program

^ | v • Reply • Share ›

GeeksforGeeks Mod ➔ Ganesh Vijay • 4 months ago

Thanks for pointing this out. We have updated the outputs.

^ | v • Reply • Share ›

Kishen Kumar Patro • 4 months ago

Beautiful Solution. Never thought of that

3 ^ | v • Reply • Share ›

-
-
-
- - [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

• Recent Comments

- [It_k](#)
 i need help for coding this function in java...
[Java Programming Language](#) · [1 hour ago](#)
- [Piyush](#)

What is the purpose of else if (recStack[*i])...

[Detect Cycle in a Directed Graph](#) · [1 hour ago](#)

- [Andy Toh](#)

My compile-time solution, which agrees with the...

[Dynamic Programming | Set 16 \(Floyd Warshall Algorithm\)](#) · [1 hour ago](#)

- [lucy](#)

because we first fill zero in first col and...

[Dynamic Programming | Set 29 \(Longest Common Substring\)](#) · [2 hours ago](#)

- [lucy](#)

@GeeksforGeeks i don't n know what is this long...

[Dynamic Programming | Set 28 \(Minimum insertions to form a palindrome\)](#) · [2 hours ago](#)

- [manish](#)

Because TAN is not a subsequence of RANT. ANT...

[Given two strings, find if first string is a subsequence of second](#) · [2 hours ago](#)

•

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

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