

Convert array into Zig-Zag fashion

Given an array of distinct elements, rearrange the elements of array in zig-zag fashion in $O(n)$ time. The converted array should be in form $a < b > c < d > e < f$.

Example:

Input: `arr[] = {4, 3, 7, 8, 6, 2, 1}`

Output: `arr[] = {3, 7, 4, 8, 2, 6, 1}`

Input: `arr[] = {1, 4, 3, 2}`

Output: `arr[] = {1, 4, 2, 3}`

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

A **Simple Solution** is to first sort the array. After sorting, exclude the first element, swap the remaining elements in pairs. (i.e. keep `arr[0]` as it is, swap `arr[1]` and `arr[2]`, swap `arr[3]` and `arr[4]`, and so on). Time complexity is $O(n \log n)$ since we need to sort the array first.

We can convert in $O(n)$ time using an **Efficient Approach**. The idea is to use modified one pass of bubble sort. Maintain a flag for representing which order (i.e. $<$ or $>$) currently we need. If the current two elements are not in that order then swap those elements otherwise not.

Let us see the main logic using three consecutive elements A, B, C. Suppose we are processing B and C currently and the current relation is ' $<$ '. But we have $B > C$. Since current relation is ' $<$ ' previous relation must be ' $>$ ' i.e., A must be greater than B. So, the relation is $A > B$ and $B > C$. We can deduce $A > C$. So if we swap B and C then the relation is $A > C$ and $C < B$. Finally we get the desired order **A C B**

Refer [this](#) for more explanation.

Below is C++ implementation of above algorithm

```
// C++ program to sort an array in Zig-Zag form
#include <iostream>
using namespace std;

// Program for zig-zag conversion of array
void zigZag(int arr[], int n)
{
    // Flag true indicates relation "<" is expected,
    // else ">" is expected. The first expected relation
```

```
// is "<"
bool flag = true;

for (int i=0; i<=n-2; i++)
{
    if (flag) /* "<" relation expected */
    {
        /* If we have a situation like A > B > C,
           we get A > B < C by swapping B and C */
        if (arr[i] > arr[i+1])
            swap(arr[i], arr[i+1]);
    }
    else /* ">" relation expected */
    {
        /* If we have a situation like A < B < C,
           we get A < C > B by swapping B and C */
        if (arr[i] < arr[i+1])
            swap(arr[i], arr[i+1]);
    }
    flag = !flag; /* flip flag */
}

// Driver program
int main()
{
    int arr[] = {4, 3, 7, 8, 6, 2, 1};
    int n = sizeof(arr)/sizeof(arr[0]);
    zigZag(arr, n);
    for (int i=0; i<n; i++)
        cout << arr[i] << " ";
    return 0;
}
```

[Run on IDE](#)

Output:

```
3 7 4 8 2 6 1
```

Time complexity: $O(n)$

Auxiliary Space: $O(1)$

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



25 Comments Category: Arrays

Related Posts:

- [Longest Span with same Sum in two Binary arrays](#)
- [Count Inversions of size three in a give array](#)
- [Find the subarray with least average](#)
- [Count triplets with sum smaller than a given value](#)
- [Find zeroes to be flipped so that number of consecutive 1's is maximized](#)
- [Reorder an array according to given indexes](#)
- [Find maximum value of Sum\(i*arr\[i\]\) with only rotations on given array allowed](#)
- [Find maximum average subarray of k length](#)

(Login to Rate and Mark)

2.5

Average Difficulty : 2.5/5.0
Based on 6 vote(s)

☐

Add to TODO List

☐

Mark as DONE

Like Share 19 people like this. Be the first of your friends.

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.

@geeksforgeeks, Some rights reserved

[Contact Us!](#)

[About Us!](#)

[Advertise with us!](#)