Rearrange a given list such that it consists of alternating minimum maximum elements

Given a list of integers, rearrange the list such that it consists of alternating minimum maximum elements **using only list operations**. The first element of the list should be minimum and second element should be maximum of all elements present in the list. Similarly, third element will be next minimum element and fourth element is next maximum element and so on. Use of extra space is not permitted.

Examples:

```
Input:  [1 3 8 2 7 5 6 4]
Output: [1 8 2 7 3 6 4 5]

Input:  [1 2 3 4 5 6 7]
Output: [1 7 2 6 3 5 4]

Input:  [1 6 2 5 3 4]
Output: [1 6 2 5 3 4]
```

The idea is to sort the list in ascending order first. Then we start popping elements from the end of the list and insert them into their correct position in the list.

Below is C++ implementation of above idea –

C/C++

```
// C++ program to rearrange a given list such that it
// consists of alternating minimum maximum elements
#include <bits/stdc++.h>
using namespace std;

// Function to rearrange a given list such that it
// consists of alternating minimum maximum elements
void alternateSort(list<int>& inp)
{
    // sort the list in ascending order
    inp.sort();

    // get iterator to first element of the list
    list<int>::iterator it = inp.begin();
    it++;

    for (int i=1; i<(inp.size() + 1)/2; i++)
    {
        // pop last element (next greatest)
        int val = inp.back();
        inp.pop_back();

        // insert it after next minimum element
        inp.insert(it, val);

        // increment the pointer for next pair
        ++it;
    }
}

// Driver code
int main()
{
    // input list
    list<int> inp({ 1, 3, 8, 2, 7, 5, 6, 4 });

    // rearrange the given list
    alternateSort(inp);

    // print the modified list
    for (int i : inp)
        cout << i << " ";

    return 0;
}
```

Java

```
// Java program to rearrange a given list such that it
// consists of alternating minimum maximum elements
import java.util.*;

class AlternateSort
{
 // Function to rearrange a given list such that it
    // consists of alternating minimum maximum elements
    // using LinkedList
 public static void alternateSort(LinkedList<Integer> ll)
 {
  Collections.sort(ll);

  for (int i = 1; i < (ll.size() + 1)/2; i++)
     {
      Integer x = ll.getLast();
      ll.removeLast();
      ll.add(2*i - 1, x);
  }

  System.out.println(ll);
 }

 public static void main (String[] args) throws java.lang.Exception
 {
  // input list
  Integer arr[] = {1, 3, 8, 2, 7, 5, 6, 4};

  // convert array to LinkedList
        LinkedList<Integer> ll = new LinkedList<Integer>(Arrays.asList(arr));

  // rearrange the given list
  alternateSort(ll);
 }
}
```

Output:

```
1 8 2 7 3 6 4 5
```