

[geeksforgeeks.org](https://www.geeksforgeeks.org)

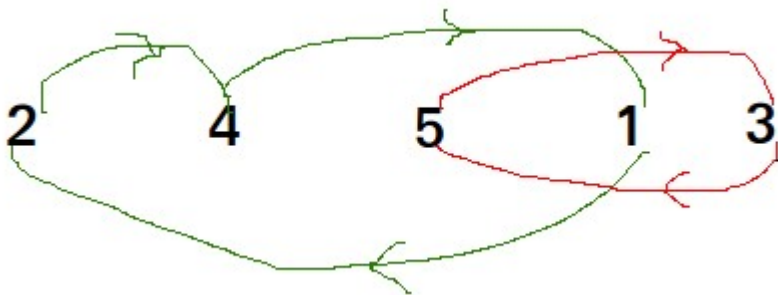
# Cycle Sort - GeeksforGeeks

6-7 minutes

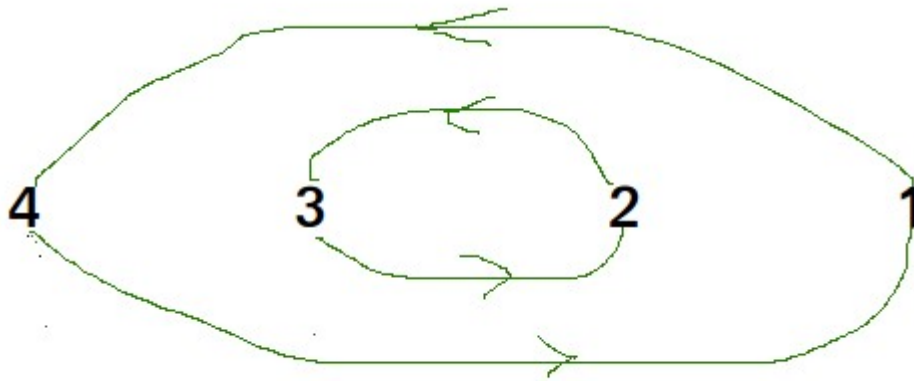
Cycle sort is an in-place sorting Algorithm, [unstable sorting algorithm](#), a comparison sort that is theoretically optimal in terms of the total number of writes to the original array.

- It is optimal in terms of number of memory writes. It [minimizes the number of memory writes](#) to sort (Each value is either written zero times, if it's already in its correct position, or written one time to its correct position.)
- It is based on the idea that array to be sorted can be divided into cycles. Cycles can be visualized as a graph. We have  $n$  nodes and an edge directed from node  $i$  to node  $j$  if the element at  $i$ -th index must be present at  $j$ -th index in the sorted array.

Cycle in  $\text{arr}[] = \{4, 5, 2, 1, 5\}$



Cycle in  $\text{arr}[] = \{4, 3, 2, 1\}$



We one by one consider all cycles. We first consider the cycle that includes first element. We find correct position of first element, place it at its correct position, say  $j$ . We consider old value of  $arr[j]$  and find its correct position, we keep doing this till all elements of current cycle are placed at correct position, i.e., we don't come back to cycle starting point.

### Explanation :

```
arr[] = {10, 5, 2, 3}
index = 0  1  2  3
cycle_start = 0
item = 10 = arr[0]
```

```
Find position where we put the item
pos = cycle_start
while (arr[i] < item)
    pos++;
```

```
We put 10 at arr[3] and change item to
old value of arr[3].
arr[] = {10, 5, 2, 10}
item = 3
```

```
Again rotate rest cycle that start with index '0'
Find position where we put the item = 3
```

```
we swap item with element at arr[1] now  
arr[] = {10, 3, 2, 10}  
item = 5
```

```
Again rotate rest cycle that start with index '0'  
and item = 5  
we swap item with element at arr[2].  
arr[] = {10, 3, 5, 10 }  
item = 2
```

```
Again rotate rest cycle that start with index '0'  
and item = 2  
arr[] = {2 ,3 , 5, 10}
```

Above is one iteration for cycle\_stat = 0.  
Repeat above steps for cycle\_start = 1, 2, ..n-2

- CPP
- Java
- Python3

## CPP

```
#include <iostream>  
  
using namespace std;  
  
void cycleSort (int arr[], int n)  
{  
  
    int writes = 0;  
  
    for (int cycle_start=0; cycle_start<=n-2;  
cycle_start++)  
  
        {
```

```
int item = arr[cycle_start];
int pos = cycle_start;
for (int i = cycle_start+1; i<n; i++)
    if (arr[i] < item)
        pos++;
if (pos == cycle_start)
    continue;
while (item == arr[pos])
    pos += 1;
if (pos != cycle_start)
{
    swap(item, arr[pos]);
    writes++;
}
while (pos != cycle_start)
{
    pos = cycle_start;
    for (int i = cycle_start+1; i<n; i++)
        if (arr[i] < item)
            pos += 1;
    while (item == arr[pos])
        pos += 1;
    if (item != arr[pos])
    {
```

```
        swap(item, arr[pos]);

        writes++;
    }

}

}

}

int main()
{
    int arr[] = {1, 8, 3, 9, 10, 10, 2, 4 };
    int n = sizeof(arr)/sizeof(arr[0]);
    cycleSort(arr, n) ;
    cout << "After sort : " << endl;
    for (int i =0; i<n; i++)
        cout << arr[i] << " ";

    return 0;
}
```

## Java

```
import java.util.*;
import java.lang.*;

class GFG
{
    public static void cycleSort (int arr[], int n)
    {
```

```
int writes = 0;

for (int cycle_start=0; cycle_start<=n-2;
cycle_start++)
{
    int item = arr[cycle_start];
    int pos = cycle_start;
    for (int i = cycle_start+1; i<n; i++)
        if (arr[i] < item)
            pos++;
    if (pos == cycle_start)
        continue;
    while (item == arr[pos])
        pos += 1;
    if (pos != cycle_start)
    {
        int temp = item;
        item = arr[pos];
        arr[pos] = temp;
        writes++;
    }
    while (pos != cycle_start)
    {
        pos = cycle_start;
        for (int i = cycle_start+1; i<n;
```

```
i++)

        if (arr[i] < item)

            pos += 1;

        while (item == arr[pos])

            pos += 1;

        if (item != arr[pos])

        {

            int temp = item;

            item = arr[pos];

            arr[pos] = temp;

            writes++;

        }

    }

}

public static void main(String[] args)

{

    int arr[] = {1, 8, 3, 9, 10, 10, 2, 4};

    int n = arr.length;

    cycleSort(arr, n) ;

    System.out.println("After sort : ");

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

        System.out.print(arr[i] + " ");

}
```

```
}
```

## Python3

```
def cycleSort(array):  
    writes = 0  
    for cycleStart in range(0, len(array) - 1):  
        item = array[cycleStart]  
        pos = cycleStart  
        for i in range(cycleStart + 1, len(array)):  
            if array[i] < item:  
                pos += 1  
        if pos == cycleStart:  
            continue  
        while item == array[pos]:  
            pos += 1  
        array[pos], item = item, array[pos]  
        writes += 1  
        while pos != cycleStart:  
            pos = cycleStart  
            for i in range(cycleStart + 1, len(array)):  
                if array[i] < item:  
                    pos += 1  
        while item == array[pos]:  
            pos += 1
```



```
        array[pos], item = item, array[pos]

    writes += 1

    return writes

arr = [1, 8, 3, 9, 10, 10, 2, 4]
n = len(arr)
cycleSort(arr)

print("After sort : ")

for i in range(0, n) :
    print(arr[i], end = ' ')
```

**Output:**

```
After sort :
1 2 3 4 8 9 10 10
```

**Time Complexity :**  $O(n^2)$ Worst Case :  $O(n^2)$ Average Case:  $O(n^2)$ Best Case :  $O(n^2)$ 

This sorting algorithm is best suited for situations where memory write or swap operations are costly.

**Reference:**

[https://en.wikipedia.org/wiki/Cycle\\_sort](https://en.wikipedia.org/wiki/Cycle_sort)

This article is contributed by [Nishant Singh](#). If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://contribute.geeksforgeeks.org) or mail your article to [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org). See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.