

1. Given an array of integers. Find the Inversion Count in the array.
2. For an array, inversion count indicates how far (or close) the array is from being sorted. If array is already sorted then the inversion count is 0. If an array is sorted in the reverse order then the inversion count is the maximum.
3. Formally, two elements $a[i]$ and $a[j]$ form an inversion if $a[i] > a[j]$ and $i < j$.

Handwritten notes for Inversion Count:

Array: 2, 4, 1, 3, 5

Indices: 0, 1, 2, 3, 4

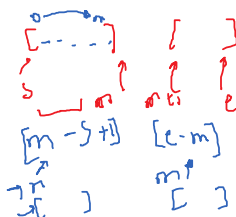
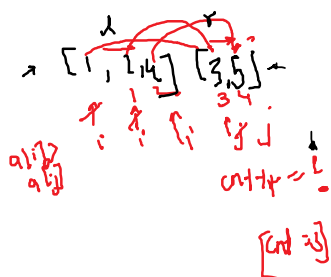
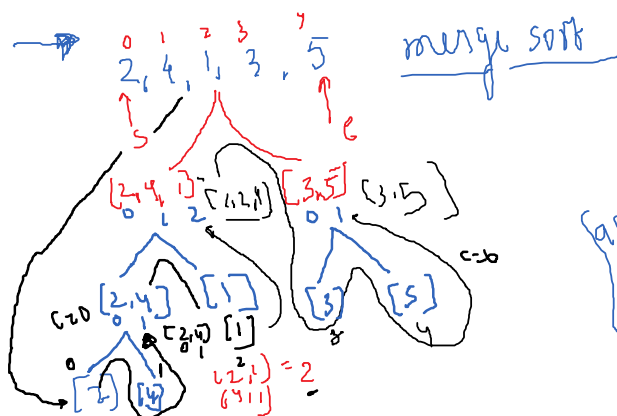
Handwritten pairs (i, j) where $a[i] > a[j]$ and $i < j$:

- (2, 1) ✓
- (4, 1) ✓
- (4, 3) ✓
- (4, 5) ✗
- (2, 3) ✗
- (2, 5) ✗
- (1, 3) ✗

for (i = 0; i < n; i++)
for (j = i + 1; j < n; j++)
if (arr[i] > arr[j]) invr++;

Time complexity: $O(n^2)$

```
int cntInversion(vector<int>&arr, int n)
{
    int invr = 0;
    for(int i = 0; i < n; i++)
    {
        for(int j = i + 1; j < n; j++)
        {
            if(arr[i] > arr[j] && i < j)
            {
                invr++;
            }
        }
    }
    return invr;
}
```

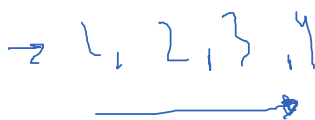


Handwritten notes: $invr = 2$, $arr = 0$, $arr = 1$.

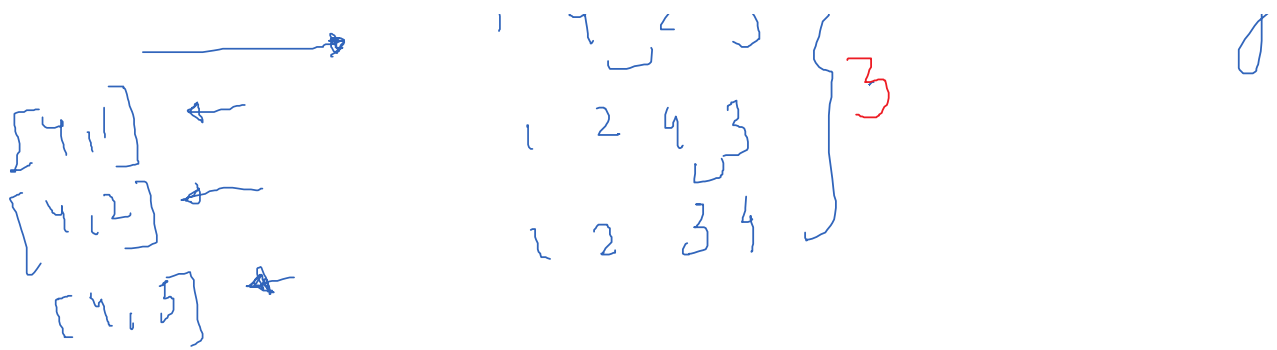
2 →

Minimum number of swaps needed

From <https://practice.geeksforgeeks.org/problems/minimum-number-of-swaps-needed2136/1#>



Handwritten notes: $a[i] > a[i+1]$, $i < j$.



3 →

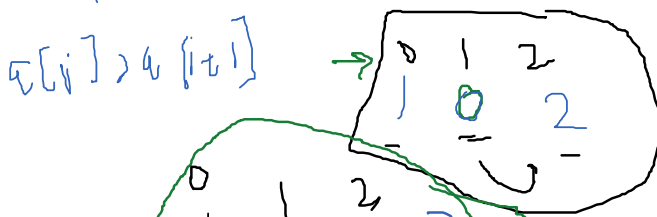
Global and Local Inversions

1, 0, 2

local or global
 $a[i] > a[i+1]$

inv min → 0 (sorted)
 inv max → n (max)

→ $\begin{matrix} 0 & 1 & 2 \\ 2 & 1 & 0 \end{matrix}$ → $\begin{matrix} (2,1) \\ (2,0) \\ (1,0) \end{matrix}$



✓ $lc=1$ True
 $g=1$

✓ $l \rightarrow l, g$



$lc = (2,0)$
 $g = (2,0) - (1,0)$

1 → 2

$0 - 1 = 1$
 $1 - 0 = 1$
 $2 - 2 = 0$

idx. val [rdx] > 1
 $0 - 1 = 1$

$1 - 2 = 1$
 $2 - 0 = 2$
 $i - a[i]$

Max → dec → 4 3 2 1

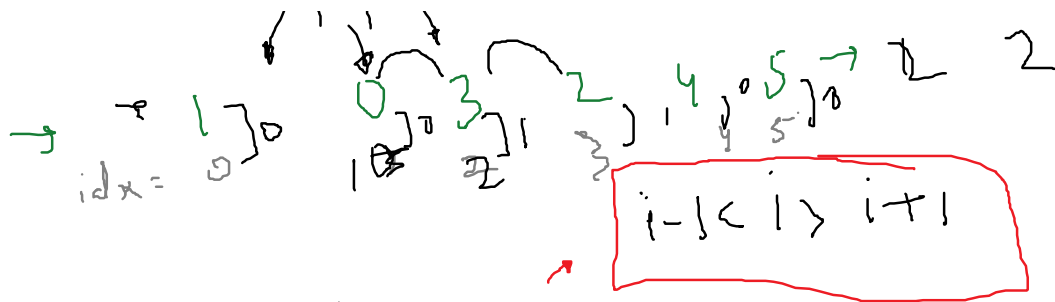
lci gi
 0 0

→ 0 → 0 → 1 → 2 → 3 →



2 2

1 2 2 4 5 → 2 2



$abs(i - a[idx]) > 1$ - false
true

arr = [9, 1, 5, 4, 8, 7, 3, 2, 6, 0]

idx = 0 1 2 3 4 5 6 7 8 9