Reversal of a String or an Array

**Problem Statement : Given a string or an array,your task is to reverse the given string or array.**

To practice this question you can click here <https://leetcode.com/problems/reverse-string/>

**Approaches to solve this question :**

Though the problem has multiple ways to solve the problem, here we will be looking at the two best approaches.

* **Iterative way**
* **Recursive way**

**Example :**

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

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

**Input : arr[] = {4, 5, 1, 2}**

**Output : arr[] = {2, 1, 5, 4}**

**Iterative way :**

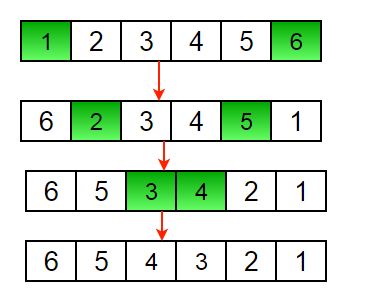
**Intuition**

To reverse a string or an array,we need to place the starting index element to the last index and last index element to the starting index.

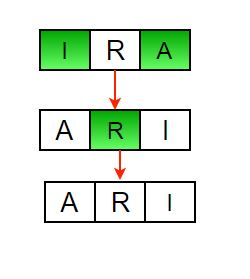
**Algorithm :**

* At first we will be initialising first and last indexes as first=0 and last=n-1,where n is the length of the given string or an array.
* Now we will be swapping arr[first] with arr[last] and now initialising first = first+1 and last = last-1,all together in a loop.

Example of an array reversal by iterative way :



Another example of String reversal be iterative way :



Code :

//JAVA

/\* Function to reverse arr[] from

first to last\*/

static void rvereseArray(int arr[],

int first, int last)

{

int tmp;

while (first < last)

{

tmp = arr[start];

arr[first] = arr[last];

arr[last] = tmp;

last--;

first++

}

}

**Time Complexity :** O(n)

**Recursive way :**

**Algorithm :**

* At first we will be initialising first and last indexes as first=0 and last=n-1,where n is the length of the given string or an array.
* Now we will be swapping arr[first] with arr[last].
* Now recursively call function reverse for the rest of the array and incrementing first to first+1 last to last-1.

Code :

//JAVA

/\* Function to reverse arr[] from start to end\*/

static void rvereseArray(int arr[], int first, int last)

{

int tmp;

if (first>=last)

return;

tmp = arr[first];

arr[first] = arr[last];

arr[last] = tmp;

rvereseArray(arr, first+1, last-1);

}

**Time Complexity :** O(n)