Write a program to reverse an array

Iterative way:

1) Initialize start and end indexes.

start = 0, end = n-1

2) In a loop, swap arr[start] with arr[end] and change start and end as follows.

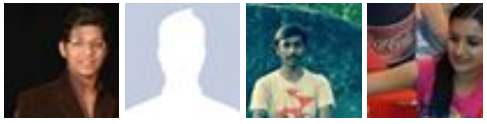
start = start +1; end = end – 1

```
/* Function to reverse arr[] from start to end*/
void rvereseArray(int arr[], int start, int end)
{
  int temp;
  while(start < end)
  {
    temp = arr[start];
    arr[start] = arr[end];
    arr[end] = temp;
    start++;
    end--;
  }
}

/* Utility that prints out an array on a line */
void printArray(int arr[], int size)
{
  int i;
  for (i=0; i < size; i++)
    printf("%d ", arr[i]);

  printf("\n");
}

/* Driver function to test above functions */
int main()
```

Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```
{
  int arr[] = {1, 2, 3, 4, 5, 6};
  printArray(arr, 6);
  rvereseArray(arr, 0, 5);
  printf("Reversed array is \n");
  printArray(arr, 6);
  getchar();
  return 0;
}
```

Time Complexity: O(n)

Recursive Way:

1) Initialize start and end indexes

start = 0, end = n-1

2) Swap arr[start] with arr[end]

3) Recursively call reverse for rest of the array.

```
/* Function to reverse arr[] from start to end*/
void rvereseArray(int arr[], int start, int end)
{
   int temp;
   if(start >= end)
     return;
   temp = arr[start];
   arr[start] = arr[end];
   arr[end] = temp;
   rvereseArray(arr, start+1, end-1);
}

/* Utility that prints out an array on a line */
void printArray(int arr[], int size)
{
  int i;
  for (i=0; i < size; i++)
    printf("%d ", arr[i]);

  printf("\n");
}

/* Driver function to test above functions */
int main()
{
  int arr[] = {1, 2, 3, 4, 5};
  printArray(arr, 5);
```

Geometric Algorithms

Popular Posts

```
  rvereseArray(arr, 0, 4);
  printf("Reversed array is \n");
  printArray(arr, 5);
  getchar();
  return 0;
}
```

Time Complexity: O(n)

Please write comments if you find any bug in the above programs or other ways to solve the same problem.

Related Tpoics:

- Remove minimum elements from either side such that 2*min becomes more than max
- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
- Find the number of zeroes
- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)

- Count all possible groups of size 2 or 3 that have sum as multiple of 3

Writing code in comment? Please use **ideone.com** and share the link here.

12 Comments GeeksforGeeks

Sort by Newest

kalyan yashwanth • 2 months ago

For example if a is an array of integers with three elements such that a[0] = 1 a

Then on reversing the array will be a[0] = 3 a[1] = 2 a[0] = 1

We copy the elements of array a into array b in reverse and then copy the arra in reverse order. We can also reverse the array without using additional memo

Please see the link for more details

http://www.fixoncloud.com/Home...

Can Eryavuz • 3 months ago

Thanks, dude!

Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 21 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 24 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 49 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 51 minutes ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

thanks for the tutorial. i like to mention a small point. in the second function, the necessary to check if start ist bigger equal end (start >= end), it's enough to c end), and the reason is bcoz we start from 0. If i miss some point over here pl Thanks again for the tutorial.

moonlight • a year ago

I don't think your 1st method will be o(n) it will be o(lg n).
you don't loop over every element.
we also could make it easier as follows:

```
int temp =0;
for(int l=0;i<n/2;i++)
{
temp=arr[l];
arr[l]=arr[n-l-1];
arr[n-l-1]=temp;
}

[sourcecode language="C"]
/* Paste your code here (You may delete these lines if not writing code) */
```


anonymous ➚ moonlight • 5 months ago

Your code is correct, but you are traversing half of the array.
The complexity of such a code is O(n/2) or O(n)
It would have been O(log n), if you were dividing the array into 2 parts a

Ujjwal ➚ moonlight • a year ago

guys is the code written by @moonlight correct.??

shek8034 ➦ Ujjwal • 11 months ago

@moonlight: Your code is absolutely correct, since we only hav
way you are doing). Because at that point, start index either cro
or they are just equal (odd length case) and all the elements ar
Same thing is done by the author in his post.
Complexity is O(n), not O(log n) [Correction for moonlight].

Ankur • 3 years ago

can u just elaborate

Anunay • 3 years ago

If there is an additional requirement where you cannot use a temp variable, the
recursive algorithm

[sourcecode language="java"]

```
//Driver
RecursiveArrayReverseWithoutTempCharacter( arr, 0, arr.Length -1, arr[0], a

public void RecursiveArrayReverseWithoutTempCharacter
(int[] arr, int start, int end, int left, int right)
{
if (start > end)
return;

arr[start++] = right;
arr[end--] = left;

RecursiveArrayReverseWithoutTempCharacter(arr, start, end, arr[start], arr[e
```

}

Note: this will work for array of any data type.

• Reply • Share ›

anonymous Anunay • 5 months ago

why so much trouble? You are passing two additional variables everyti
extra space.
There is a pretty standard to swap without using the third (temp) variab

1.) a=a*b;
2.) b=a/b;
3.) a=a/b;

Note: This will work for both positive and negative integers. But for larg
which case, either switch to long long or use + instead of * and - instea

• Reply • Share ›

sharat Anunay • 3 years ago

If temp is not to be used, use the XOR mechanism to swap, not this w

This is more inefficient than using temp, This will two extra variables to
new function stack corresponding to a recursive call.)

• Reply • Share ›

neham sharat • a year ago

To swap two variables w/o using third can be done by simple a
a = a + b;
b = a - b;
a = a - b;

```
/* Paste your code here (You may delete these lines if r
```