Left Rotation ☆

1337.53 more points to get your next star!

Rank: **138221** | Points: **862.47/2200**

Problem Solving ★★★★★

Problem Submissions Leaderboard Discussions **Editorial**

Editorial by AllisonP

To solve this challenge, we perform the following basic steps:

1. Create a new n-element (where n is the length of arr) array named $rotated$ to hold the rotated items.
2. Copy the contents of arr over to the new array in two parts:
   - The d-element contiguous segment from arr_0 to arr_{d-1} must be copied over to the contiguous segment starting at $rotated_{n-d}$ and ending at $rotated_{n-1}$.
   - The $(n-d)$-element contiguous segment from arr_d to arr_{n-1} must be copied over to the contiguous segment starting at $rotated_0$ and ending at $rotated_d$.
3. Reassign the reference to arr so that it points to $rotated$ instead.

This is implemented by the following Java code:

```
public static int[] rotateArray(int[] arr, int d){
    // Because the constraints state d < n, we need not concern ourselves with shifting > n u
    int n = arr.length;

    // Create new array for rotated elements:
    int[] rotated = new int[n];

    // Copy segments of shifted elements to rotated array:
    System.arraycopy(arr, d, rotated, 0, n - d);
    System.arraycopy(arr, 0, rotated, n - d, d);

    return rotated;
}
```

Set by jpierce88

Problem Setter's code:

Python 3

```
def rotateLeft(d, arr):
    return arr[d:]+arr[:d]
```

Tested by AllisonP

Problem Tester's code:

Java

```
import java.util.*;

public class Solution {

    public static int[] rotateArray(int[] arr, int d){
        // Because the constraints state d < n, we need not concern ourselves with shifting >
        int n = arr.length;

        // Create a temporary d-element array to store elements shifted to the left of index
        int[] temp_arr = Arrays.copyOfRange(arr, 0, d);

        // Shift elements from indices d through n to indices 0 through d - 1:
        for(int i = d; i < n; i++) {
            arr[i - d] = arr[i];
        }

        // Copy the d shifted elements from the temporary array back to the original array:
        for(int i = n - d; i < n; i++) {
            arr[i] = temp_arr[i-n+d];
        }
```

STATISTICS

Difficulty: Easy

Time Complexity:

Publish Date: Jul 06 2016

This is a Practice Challenge

NEED HELP?

View discussions

View top submissions