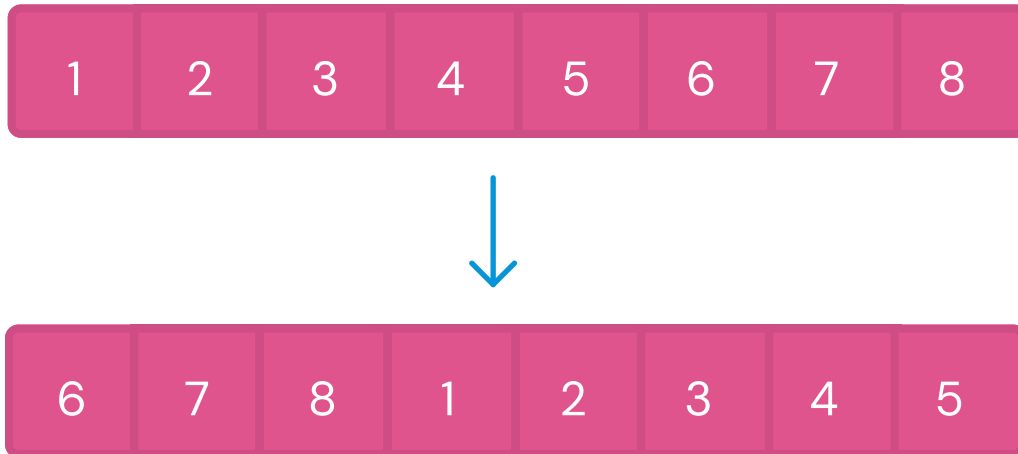


ARRAY ROTATIONS

Rotated array

An array is said to be n times rotated if all its elements are shifted by n positions.

Example: Here is an array with length 8 rotated by 3 elements



Write a function to rotate given array by n elements

This problem can be solved by 4 methods

- using temp array
- rotating one by one for d times
- reversal algorithm
- block swap algorithm

Method 1 (using temp array)

let us assume input to be `arr=[2,3,4,8,6,8,9,10]` and $d=3$.

Step 1- copy first d elements in a temp array

`temp = [2,3,4]`

Step 2- shift all elements by d elements;

from $i=0$ to $i=arr.length$

`arr[i] = a[i+d];`

we get `arr = [8,6,8,9,10,8,9,10]`

Step 3- put temp array elements back in arr

`arr = [8,6,8,9,10,2,3,4]`

Time complexity

we used three loops two loops d times and one loop n times. So time complexity will be
 $\Rightarrow O(n+d+d)$
 $\Rightarrow O(n+2d)$

Auxiliary Space

we used temp array of length d to save element. So Auxiliary Space is $O(d)$