

# Vector Rotation

The Problem  
Three Algorithms  
Implementations

## The Problem

### The Problem

Rotate vector  $x[n]$  left by  $d$  positions.

For  $n=8$  and  $d=3$ , change *abcdefgh* to *defghabc*.

Constraints:  $O(n)$  time,  $O(1)$  extra space.

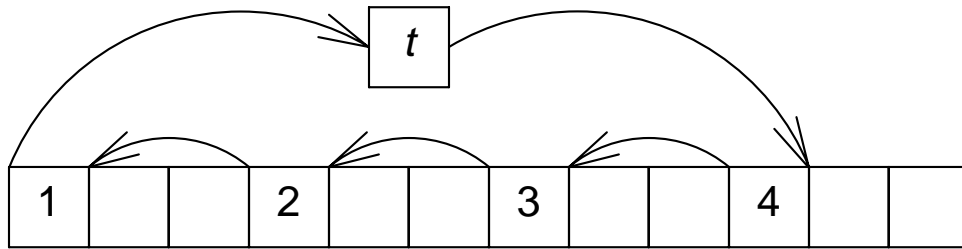
### Pricey Solutions

Store  $d$  in intermediate vector, shift the rest, store back. [ $O(n)$  extra space.]

Rotate by 1  $d$  times. [ $O(n)$  time.]

# A Juggling Algorithm

The Idea ( $n = 12$ ,  $d = 3$ )



## The Code

```
for i = [0, gcd(d, n))
    /* move i-th values of blocks */
    t = x[i]
    j = i
    loop
        k = j + d
        if k >= n
            k -= n
        if k == i
            break
        x[j] = x[k]
        j = k
    x[j] = t
```

## The Block-Swap Algorithm

The Idea: Change  $ab$  to  $ba$

If  $a$  is shorter, divide  $b$  into  $b_l$  and  $b_r$ .

Swap  $a$  and  $b_r$  to change  $ab_l b_r$  into  $b_r b_l a$ .

Recur on pieces of  $b$ .

The Code

```
if d == 0 || d == n
    return
i = p = d
j = n - p
while i != j
    if i > j
        swap(p-i, p, j)
        i -= j
    else
        swap(p-i, p+j-i, i)
        j -= i
swap(p-i, p, i)
```

# The Reversal Algorithm

## The Idea

Reverse  $a$  to get  $a^r b$ .

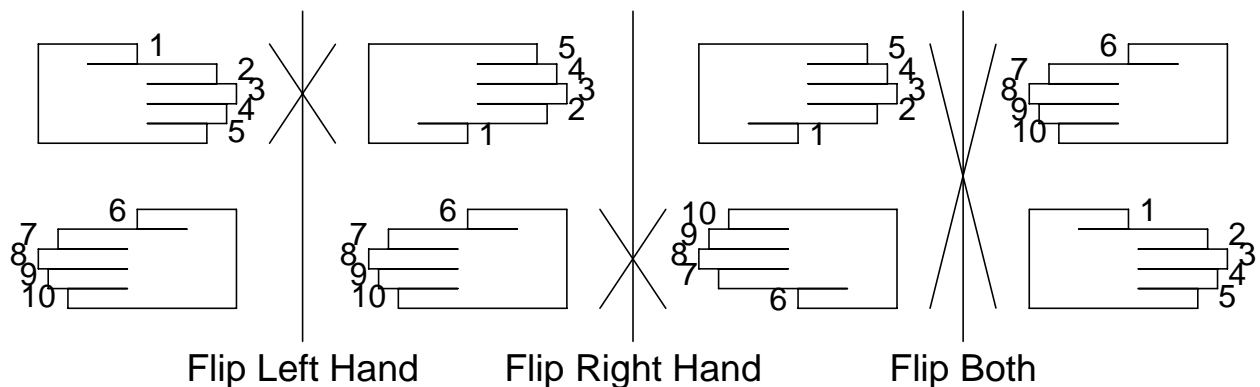
Reverse  $b$  to get  $a^r b^r$ .

Reverse all to get  $(a^r b^r)^r = ba$ .

The Code `/* rotate abcdefgh left three */`

```
reverse(0, d-1)      /* cbadefgh */
reverse(d, n-1)      /* cbahgfed */
reverse(0, n-1)      /* defghabc */
```

## Doug McIlroy's Handwaving Description



## An Experiment on Run Times

$n = 10^6$ , 400MHz Pentium II.

