

绪论

局限：缓存

不学诗，何以言；不学礼，何以立

He has given signs of himself which are visible to those who seek him, and not to those who do not seek him.

邓俊辉

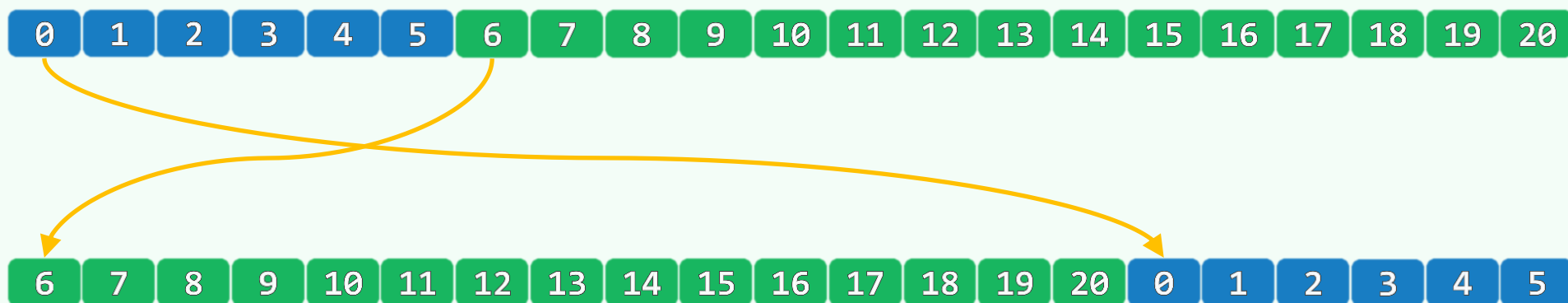
deng@tsinghua.edu.cn

# 就地循环位移

❖ //仅用 $O(1)$ 辅助空间，将数组 $A[0, n)$ 中的元素向左循环移动 $k$ 个单元

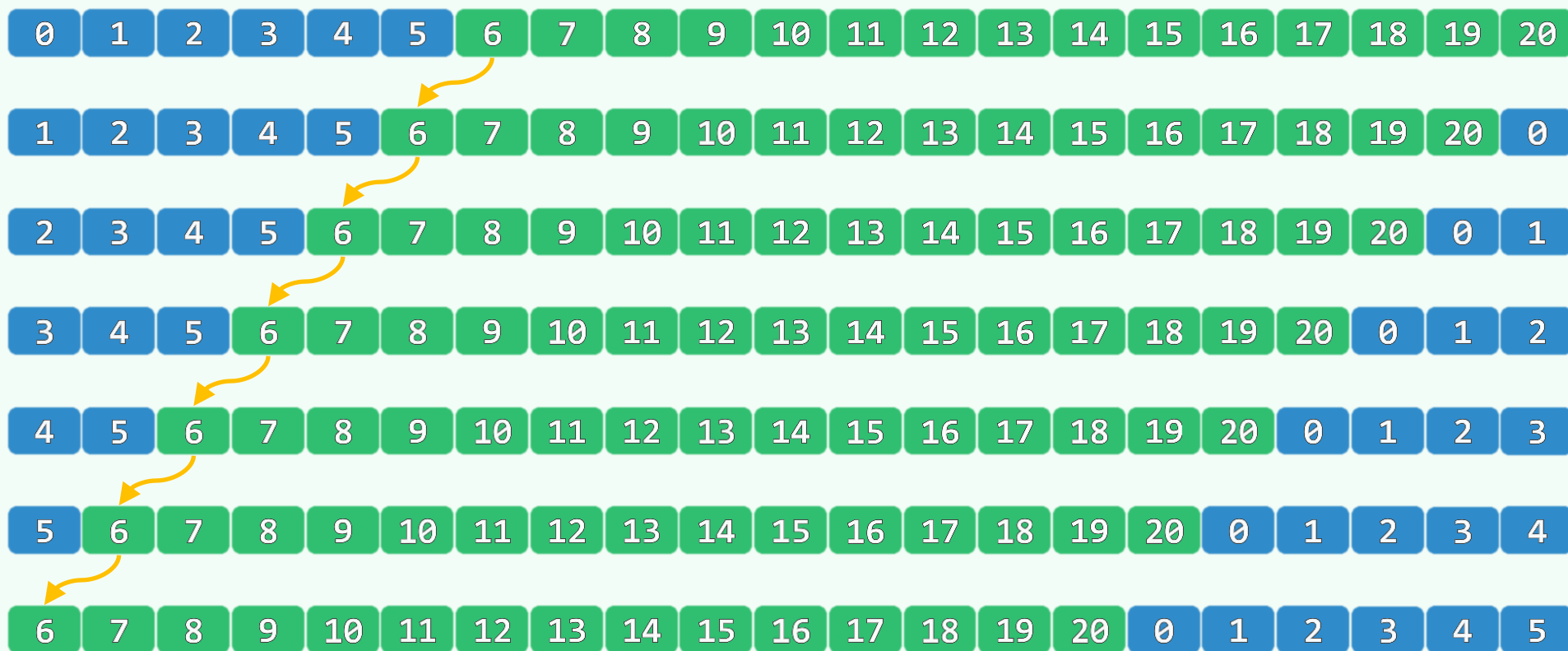
```
void shift( int * A, int n, int k );
```

❖ 比如：shift( A, 21, 6 );



# 蛮力版

❖ `void shift0( int * A, int n, int k )` //反复以1为间距循环左移  
`{ while ( k-- ) shift( A, n, 0, 1 ); }` //共迭代k次,  $O(n*k)$



# 迭代版

```
❖ int shift( int * A, int n, int s, int k ) { //  $O(n / \text{GCD}(n, k))$ 

    int b = A[s]; int i = s, j = (s + k) % n; int mov = 0; //mov记录移动次数

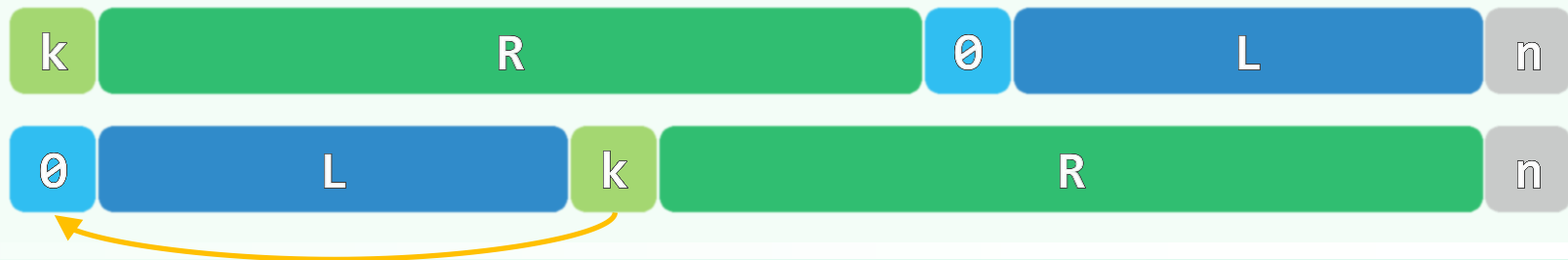
    while ( s != j ) //从A[s]出发, 以k为间隔, 依次左移k位

        { A[i] = A[j]; i = j; j = (j + k) % n; mov++; }

    A[i] = b; return mov + 1; //最后, 起始元素转入对应位置

} //  $[0, n)$ 由关于k的 $g = \text{GCD}(n, k)$ 个同余类组成, shift(s, k)能够且只能够使其中之一就位
```

❖ 其它的同余类呢...



# 迭代版

```
void shift1(int* A, int n, int k) { //经多轮迭代，实现数组循环左移k位，累计 $O(n+g)$ 
```

```
    for (int s = 0, mov = 0; mov < n; s++) // $O(g) = O(\text{GCD}(n, k))$ 
```

```
        mov += shift(A, n, s, k);
```

```
}
```



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
6	1	2	9	4	5	12	7	8	15	10	11	18	13	14	0	16	17	3	19	20
6	7	2	9	10	5	12	13	8	15	16	11	18	19	14	0	1	17	3	4	20
6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	0	1	2	3	4	5

# 倒置版

// 借助倒置算法，将数组循环左移k位

```
void shift2( int * A, int n, int k ) {  
  
    reverse( A, k ); //  $O(3k/2)$   
  
    reverse( A + k, n - k ); //  $O(3(n-k)/2)$   
  
    reverse( A, n ); //  $O(3n/2)$   
  
} //  $O(3n)$ 
```

