

不学诗，何以言
不学礼，何以立

1. 绪论

局限

缓存

邓俊辉

deng@tsinghua.edu.cn

循环位移

❖ //将数组A[0, n)中元素向左循环移动k个单元

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

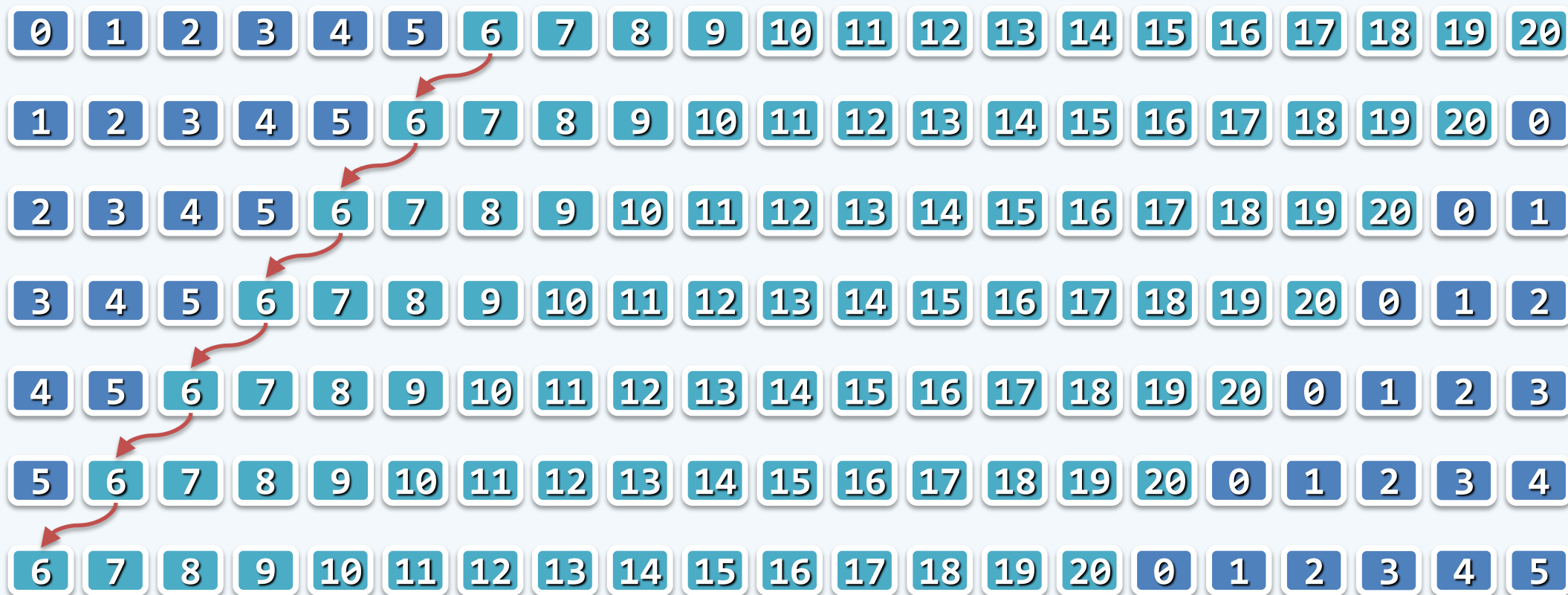
❖ 比如 : shift(A, 21, 6);

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

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

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, $O(nk)$



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

//各含 n/g 个元素

$\text{shift}(s, k)$ 能够且只能够使其中之一就位

//即 s 所属的同余类

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

```
void shift2( int * A, int n, int k ) { //借助倒置算法，将数组循环左移k位， $O(3n)$   
    reverse( A, k ); // $O(3k/2)$   
    reverse( A + k, n - k ); // $O(3(n-k)/2)$   
    reverse( A, n ); // $O(3n/2)$   
}
```



0 1 2 3 4 5 6 7 8

2 1 0 8 7 6 5 4 3

3 4 5 6 7 8 0 1 2