

排序

希尔排序：框架+实例

14-C1

邓俊辉

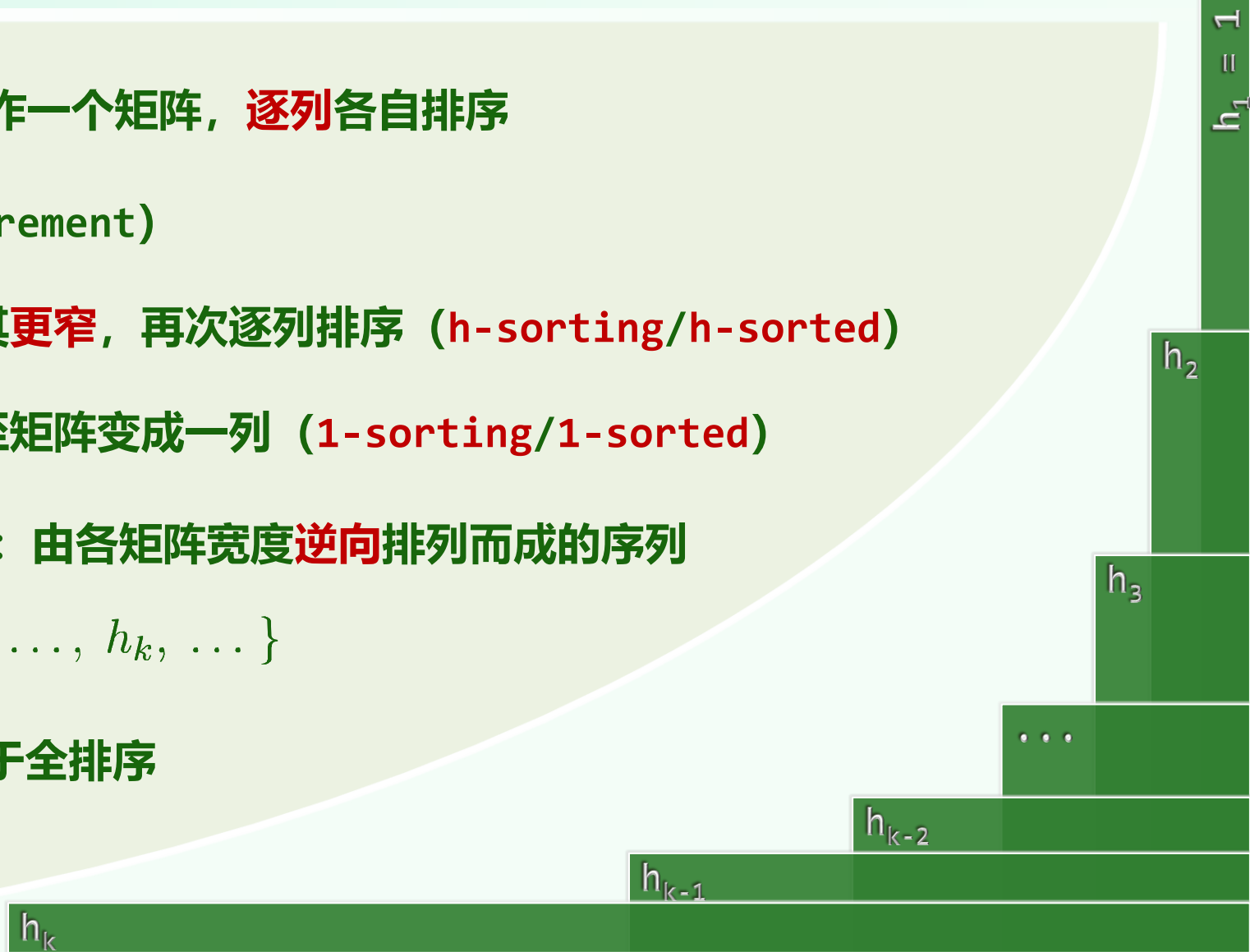
deng@tsinghua.edu.cn

瓜熟蒂落，水到渠成

Shellsort

- ❖ D. L. Shell: 将整个序列视作一个矩阵, **逐列**各自排序
- ❖ 递减增量 (diminishing increment)
 - 由粗到细: 重排矩阵, 使其**更窄**, 再次逐列排序 (**h-sorting/h-sorted**)
 - 逐步求精: 如此往复, 直至矩阵变成一列 (**1-sorting/1-sorted**)
- ❖ 步长序列 (step sequence) : 由各矩阵宽度**逆向**排列而成的序列
- ❖ 正确性: 最后一次迭代, 等同于全排序

1-sorted = ordered



实例: $h_5 = 8$

80 23 19 40 85 1 18 92 71 8 96 46 12

80	23	19	40	85	1	18	92
71	8	96	46	12			

71	8	19	40	12	1	18	92
80	23	96	46	85			

71 8 19 40 12 1 18 92 80 23 96 46 85

实例: $h_4 = 5$

1 8 19 40 12 71 18 85 80 23 96 46 92

71 8 19 40 12
1 18 92 80 23
96 46 85

1 8 19 40 12
71 18 85 80 23
96 46 92

71 8 19 40 12 1 18 92 80 23 96 46 85

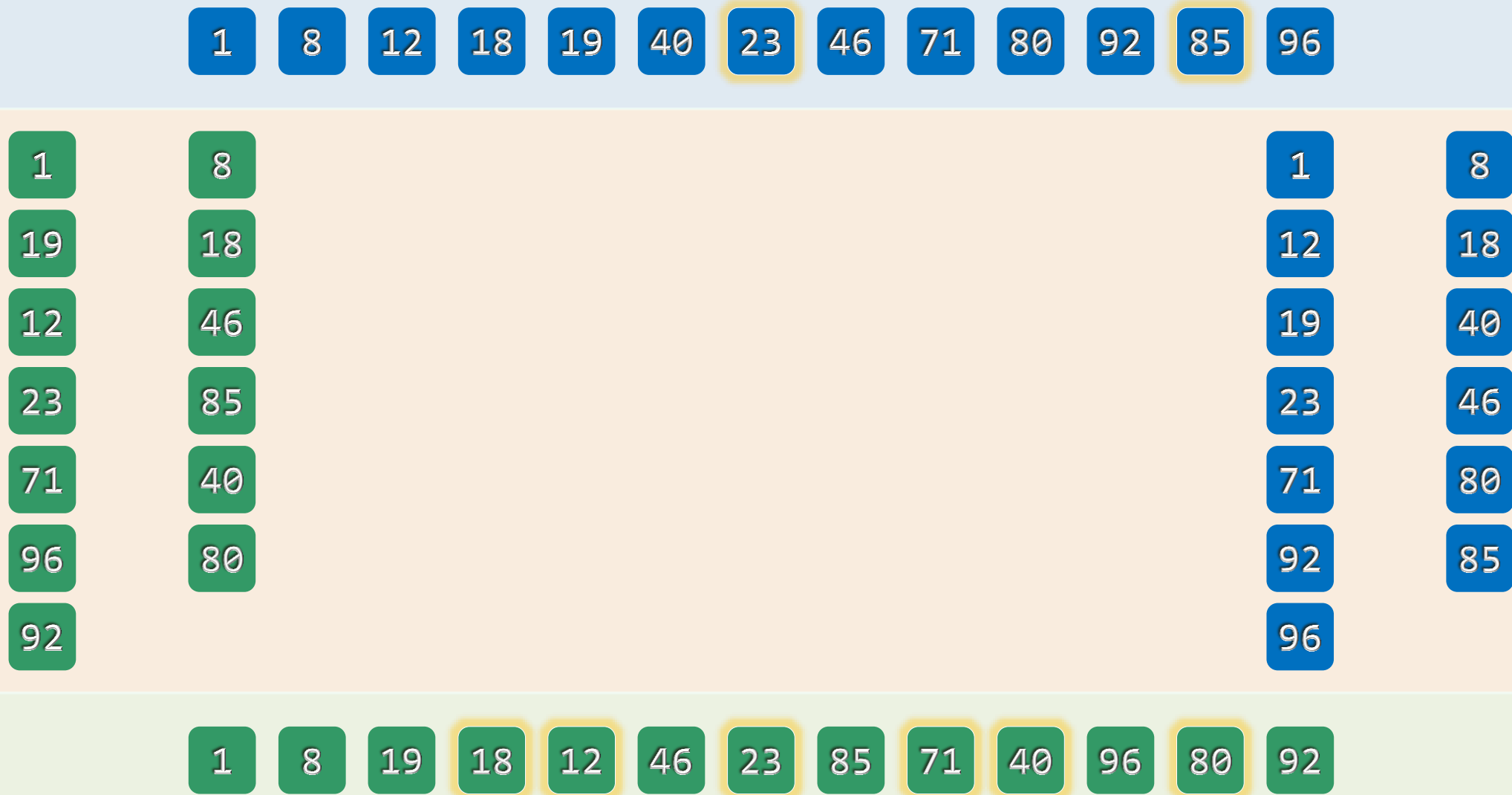
实例: $h_3 = 3$

1 8 19 40 12 71 18 85 80 23 96 46 92

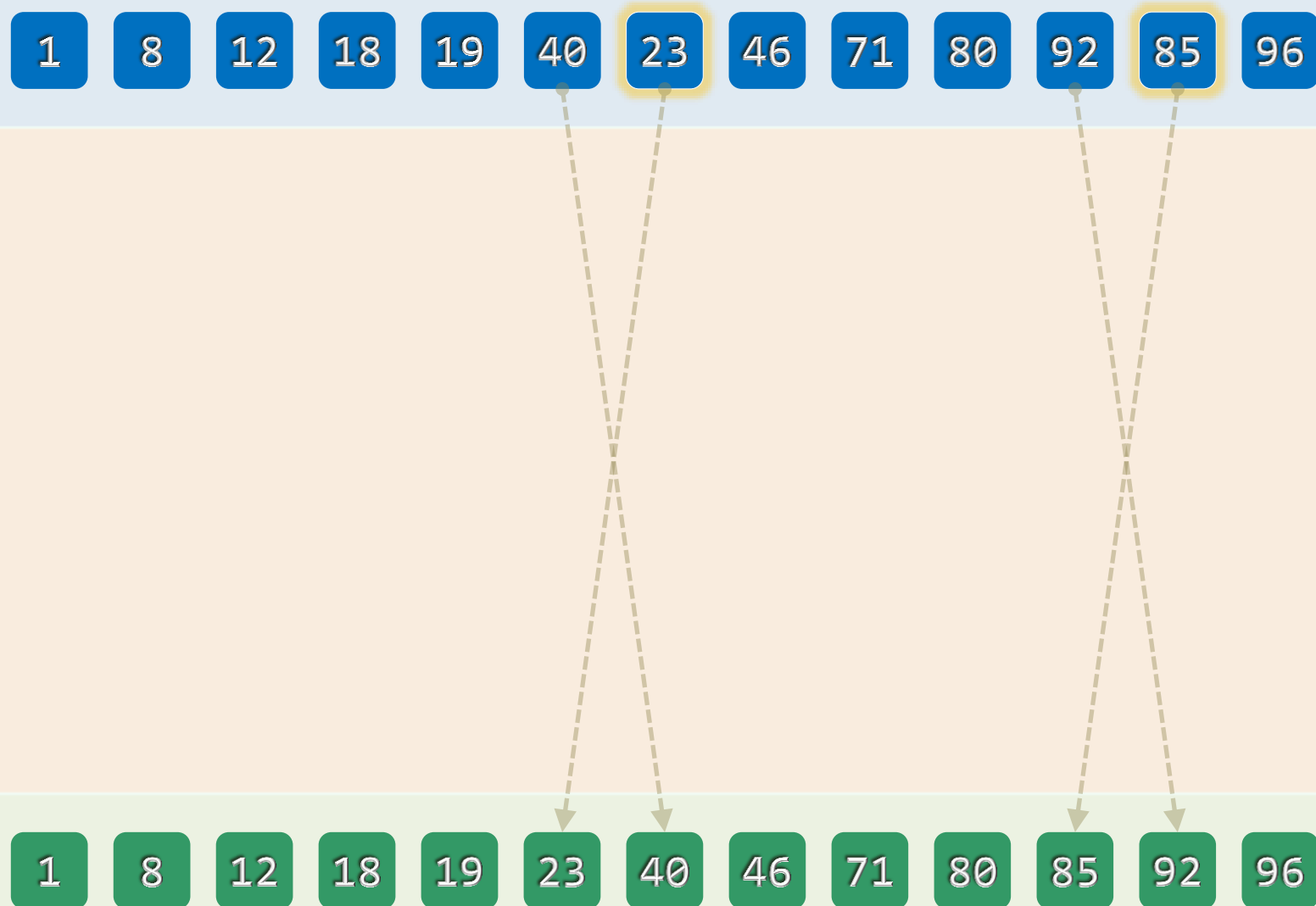
1	8	19		1	8	19
40	12	71		18	12	46
18	85	80		23	85	71
23	96	46		40	96	80
92				92		

1 8 19 18 12 46 23 85 71 40 96 80 92

实例: $h_2 = 2$



实例: $h_1 = 1$



Call-by-rank

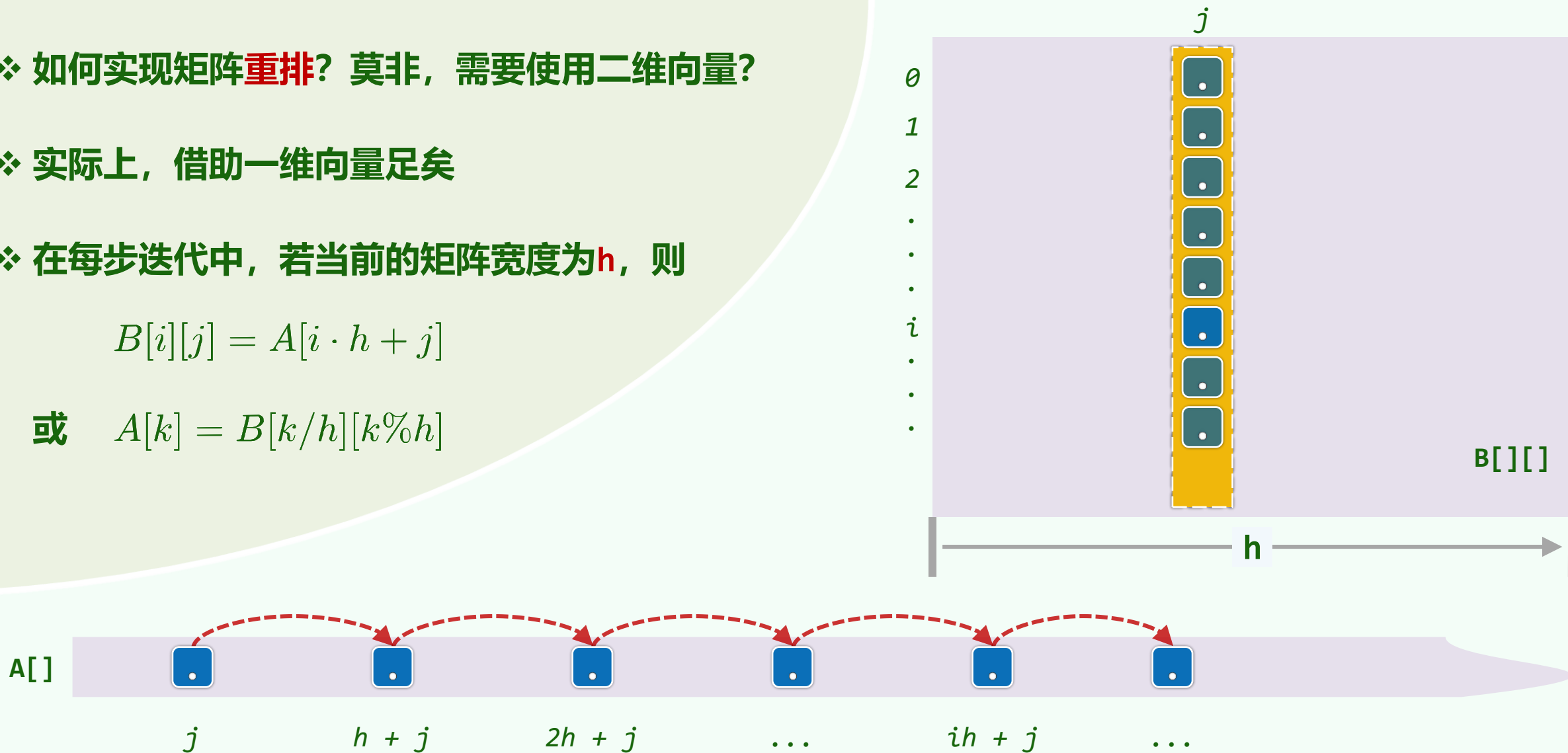
❖ 如何实现矩阵**重排**? 莫非, 需要使用二维向量?

❖ 实际上, 借助一维向量足矣

❖ 在每步迭代中, 若当前的矩阵宽度为**h**, 则

$$B[i][j] = A[i \cdot h + j]$$

或 $A[k] = B[k/h][k\%h]$



实现

```
template <typename T> void Vector<T>::shellSort( Rank lo, Rank hi ) {  
    // Using PS Sequence { 1, 3, 7, 15, 31, 63, 127, ..., 1073741823, ... }  
    for ( Rank d = 0x3FFFFFFF; 0 < d; d >>= 1 )  
        for ( Rank j = lo + d; j < hi; j++ ) { //for each j in [lo+d, hi)  
            T x = _elem[j]; Rank i = j - d;  
            while ( lo <= i && _elem[i] > x )  
                { _elem[i + d] = _elem[i]; i -= d; }  
            _elem[i + d] = x; //insert [j] into its subsequence  
        }  
    }  
} //0 <= lo < hi <= size <= 2^30
```