

本节内容

处理冲突的 方法

——拉链法

知识总览

处理冲突的方法 ——拉链法

如何插入一个元素

如何查找一个元素

—— 学会分析查找长度

如何删除一个元素

Tips: 散列表通常不考代码，着重掌握手算分析方法

散列表的插入操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(\text{key}) = \text{key} \% 13$ ，用拉链法解决冲突。依次插入关键字 {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}

0	1	2	3	4	5	6	7	8	9	10	11	12
^	^	^	^	^	^	^	^	^	^	^	^	^

可以视为
13个链表的
头指针

$19 \% 13 = 6$	$27 \% 13 = 1$
$14 \% 13 = 1$	$55 \% 13 = 3$
$23 \% 13 = 10$	$11 \% 13 = 11$
$1 \% 13 = 1$	$10 \% 13 = 10$
$68 \% 13 = 3$	$79 \% 13 = 1$
$20 \% 13 = 7$	
$84 \% 13 = 6$	

拉链法（又称链接法、链地址法）：把所有“同义词”存储在一个链表中

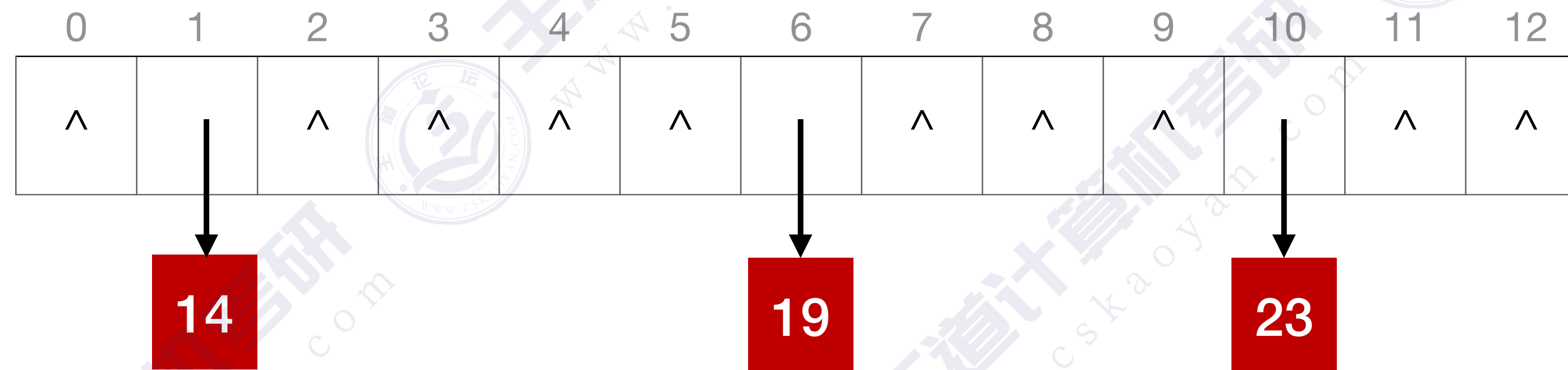
如何在散列表（拉链法解决冲突）中插入一个新元素？

Step 1: 结合散列函数计算新元素的散列地址

Step 2: 将新元素插入散列地址对应的链表（可用头插法，也可用尾插法）

散列表的插入操作（拉链法解决冲突）

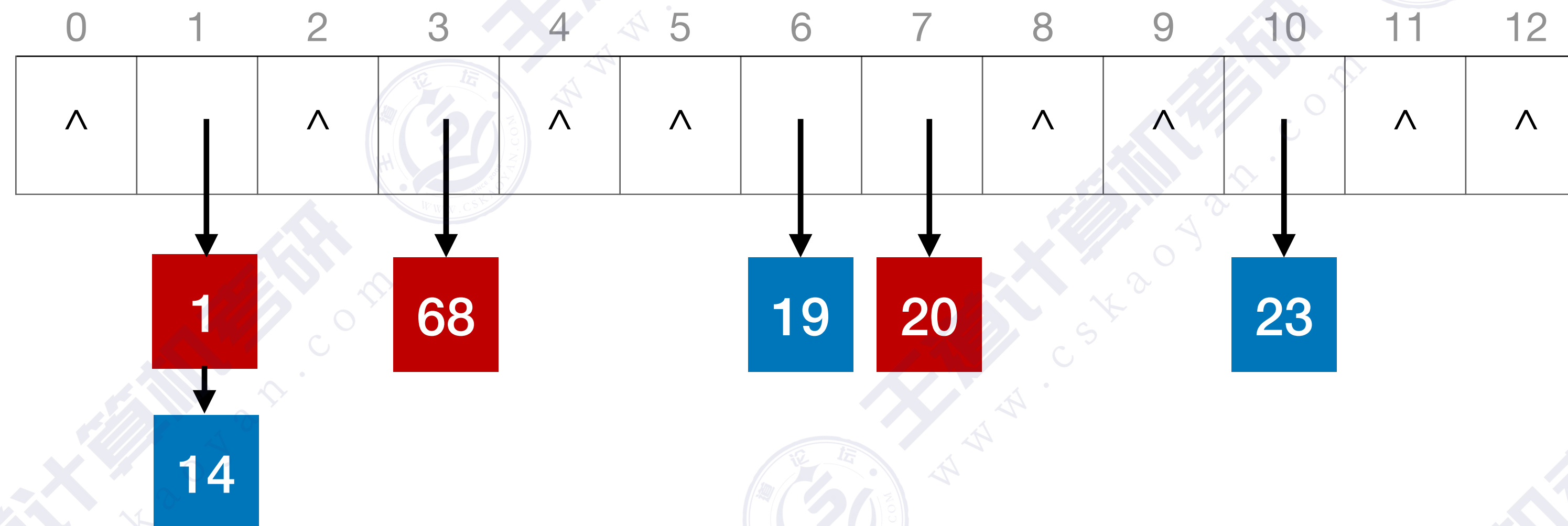
例：若散列表长度为13，散列函数 $H(\text{key}) = \text{key} \% 13$ ，用拉链法解决冲突。依次插入关键字 {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}



$19 \% 13 = 6$	$27 \% 13 = 1$
$14 \% 13 = 1$	$55 \% 13 = 3$
$23 \% 13 = 10$	$11 \% 13 = 11$
$1 \% 13 = 1$	$10 \% 13 = 10$
$68 \% 13 = 3$	$79 \% 13 = 1$
$20 \% 13 = 7$	
$84 \% 13 = 6$	

散列表的插入操作（拉链法解决冲突）

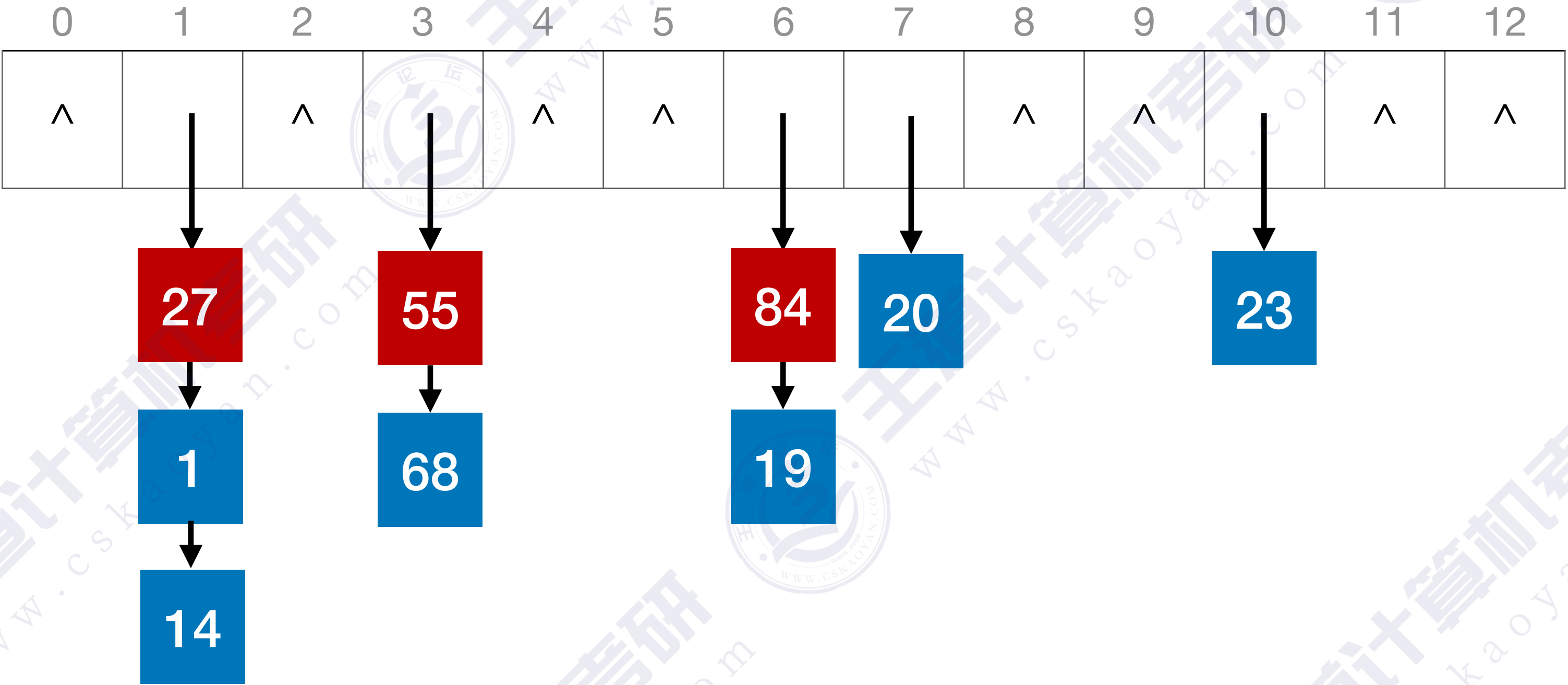
例：若散列表长度为13，散列函数 $H(\text{key}) = \text{key} \% 13$ ，用拉链法解决冲突。依次插入关键字 {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}



$19 \% 13 = 6$	$27 \% 13 = 1$
$14 \% 13 = 1$	$55 \% 13 = 3$
$23 \% 13 = 10$	$11 \% 13 = 11$
$1 \% 13 = 1$	$10 \% 13 = 10$
$68 \% 13 = 3$	$79 \% 13 = 1$
$20 \% 13 = 7$	
$84 \% 13 = 6$	

散列表的插入操作（拉链法解决冲突）

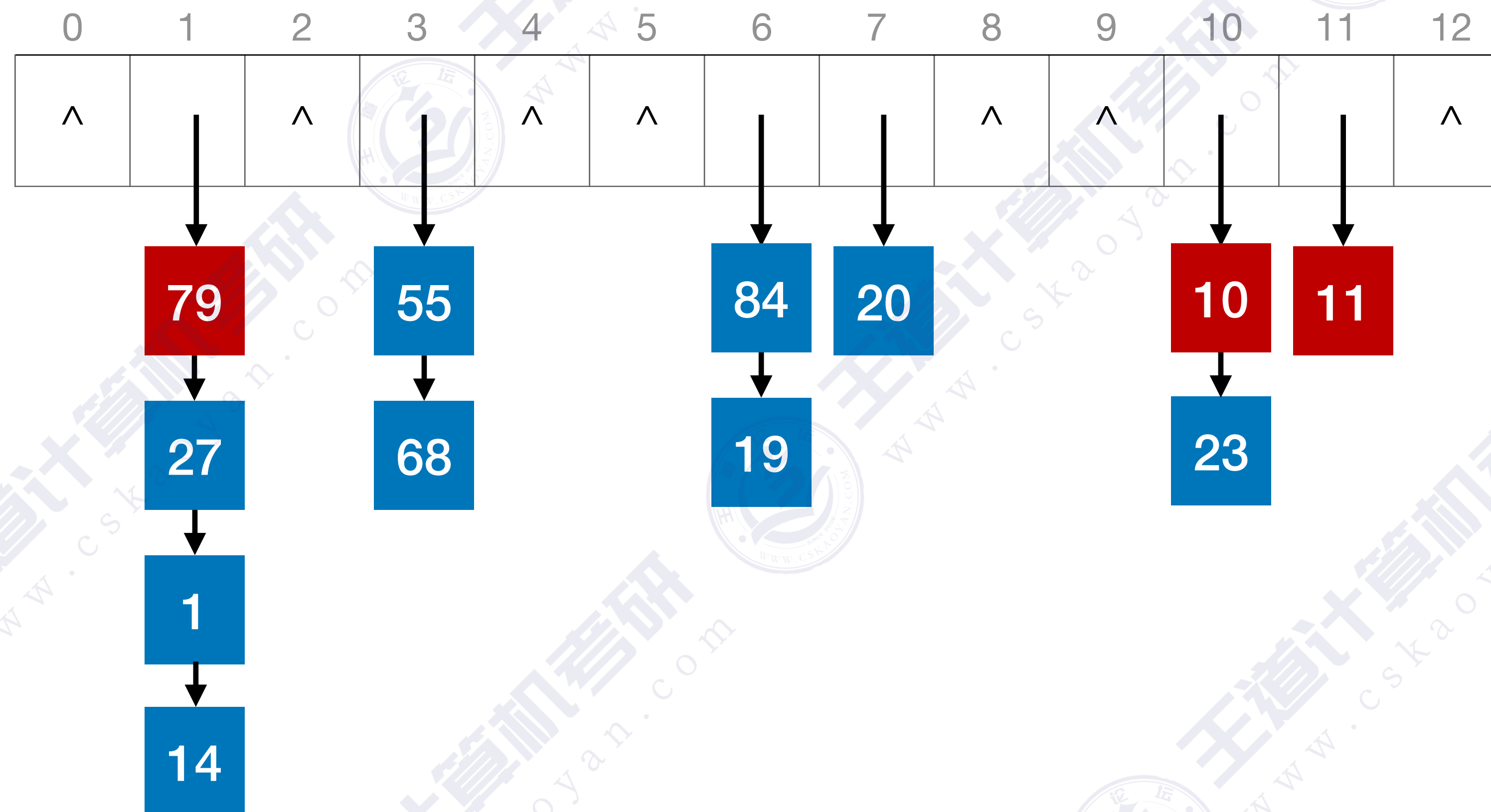
例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。依次插入关键字 {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}



$19\%13=6$	$27\%13=1$
$14\%13=1$	$55\%13=3$
$23\%13=10$	$11\%13=11$
$1\%13=1$	$10\%13=10$
$68\%13=3$	$79\%13=1$
$20\%13=7$	
$84\%13=6$	

散列表的插入操作（拉链法解决冲突）

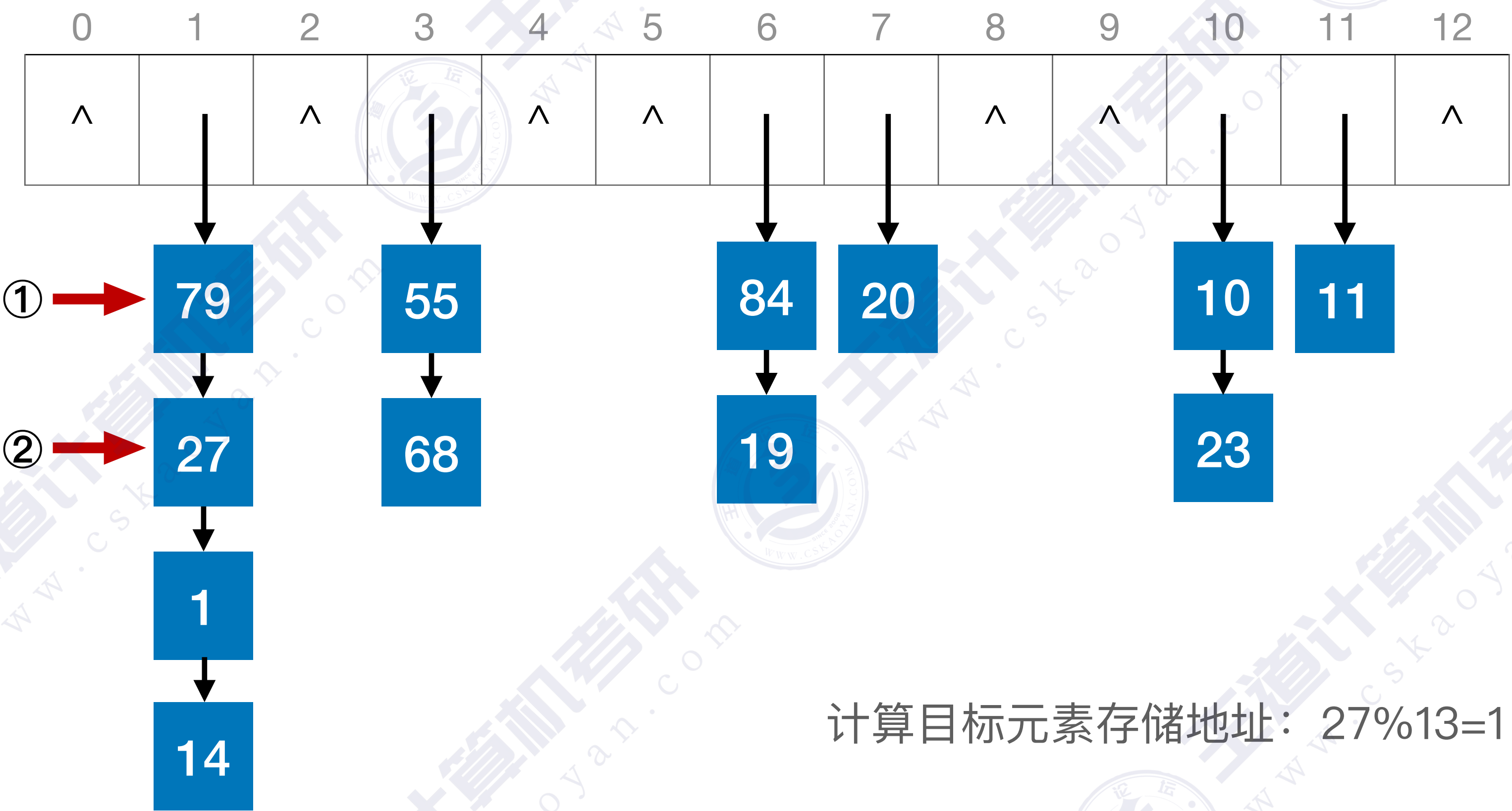
例：若散列表长度为13，散列函数 $H(\text{key}) = \text{key} \% 13$ ，用拉链法解决冲突。依次插入关键字 {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}



$19 \% 13 = 6$	$27 \% 13 = 1$
$14 \% 13 = 1$	$55 \% 13 = 3$
$23 \% 13 = 10$	$11 \% 13 = 11$
$1 \% 13 = 1$	$10 \% 13 = 10$
$68 \% 13 = 3$	$79 \% 13 = 1$
$20 \% 13 = 7$	
$84 \% 13 = 6$	

散列表的查找操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。在下述散列表中依次查找元素 27，20，66，21，分析每次查找操作的过程和查找长度。



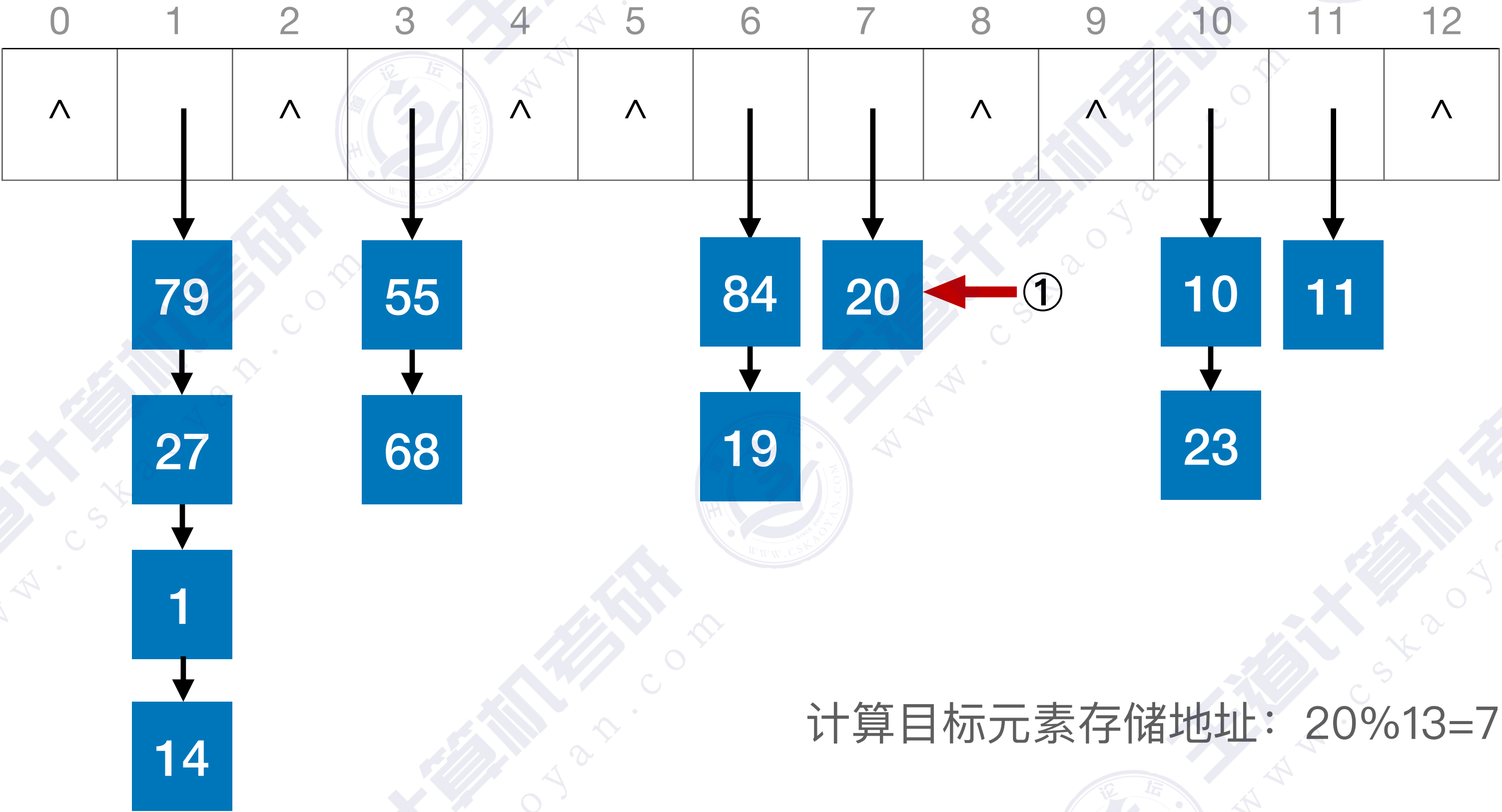
查找目标：27

计算目标元素存储地址： $27\%13=1$

27查找成功，查找长度=2

散列表的查找操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。在下述散列表中依次查找元素 27，20，66，21，分析每次查找操作的过程和查找长度。



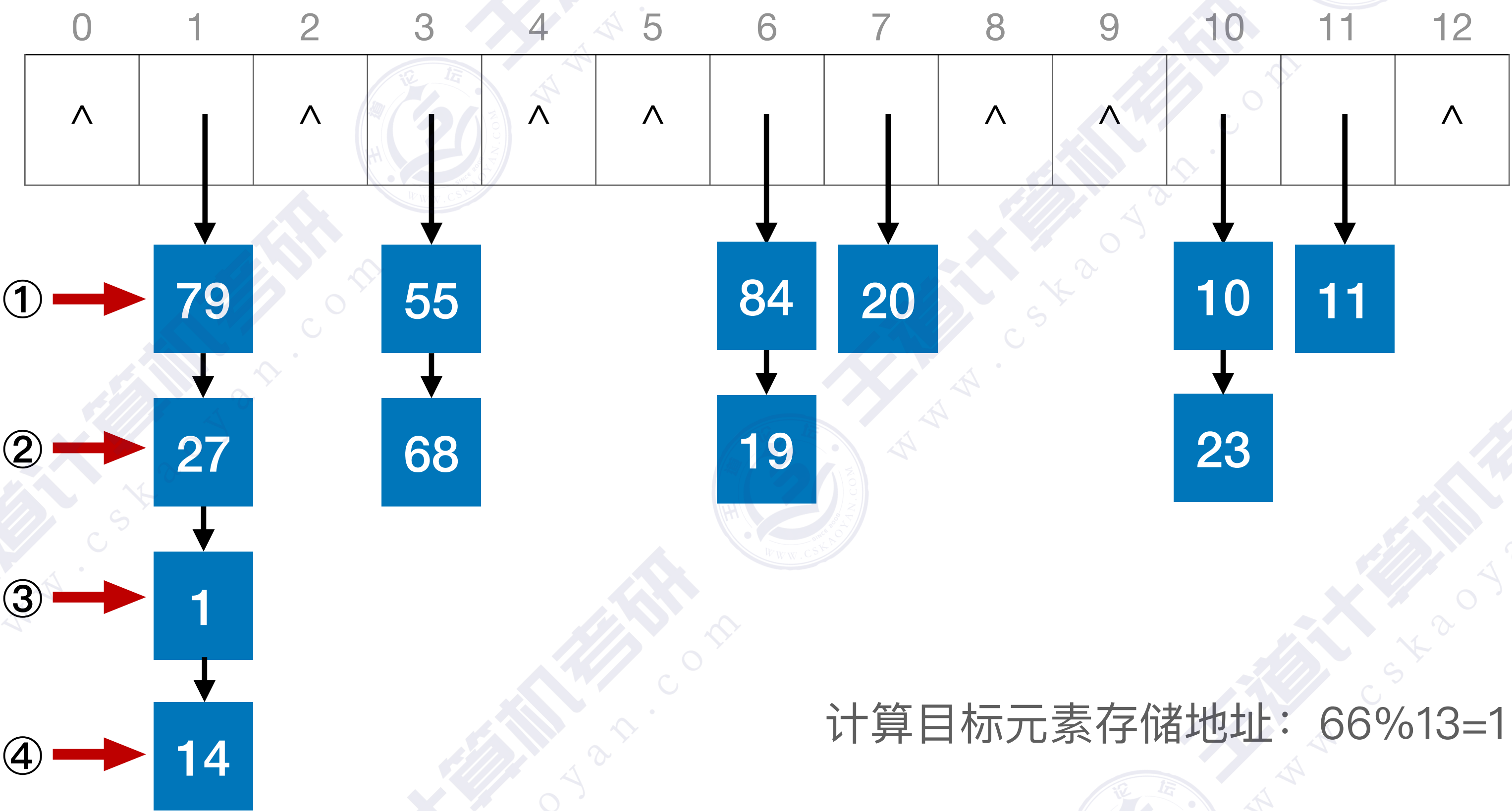
查找目标：20

计算目标元素存储地址： $20\%13=7$

20查找成功，查找长度=1

散列表的查找操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。在下述散列表中依次查找元素 27，20，66，21，分析每次查找操作的过程和查找长度。



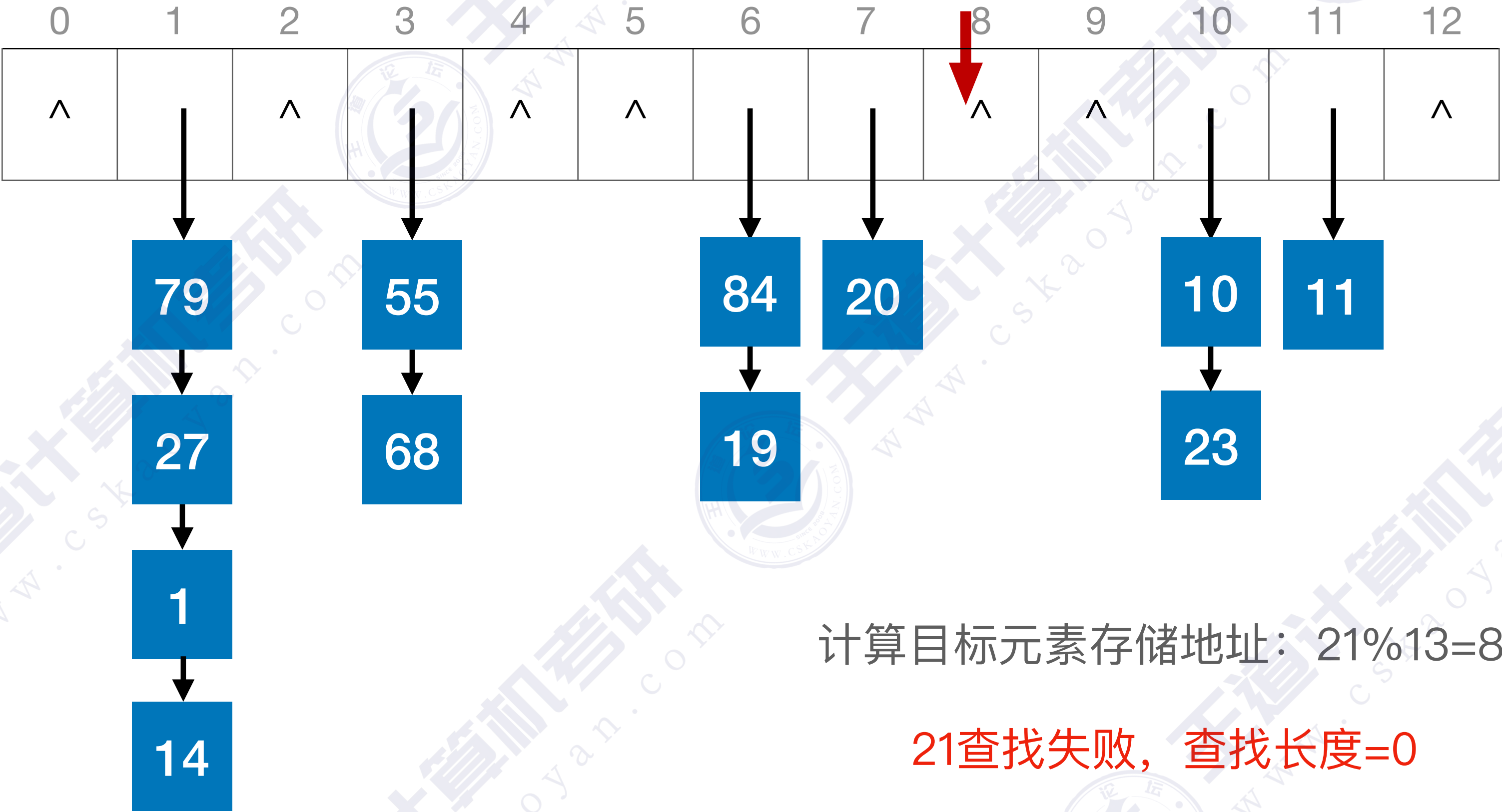
查找目标：66

计算目标元素存储地址： $66\%13=1$

66查找失败，查找长度=4

散列表的查找操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。在下述散列表中依次查找元素 27，20，66，21，分析每次查找操作的过程和查找长度。



查找目标：21

计算目标元素存储地址： $21\%13=8$

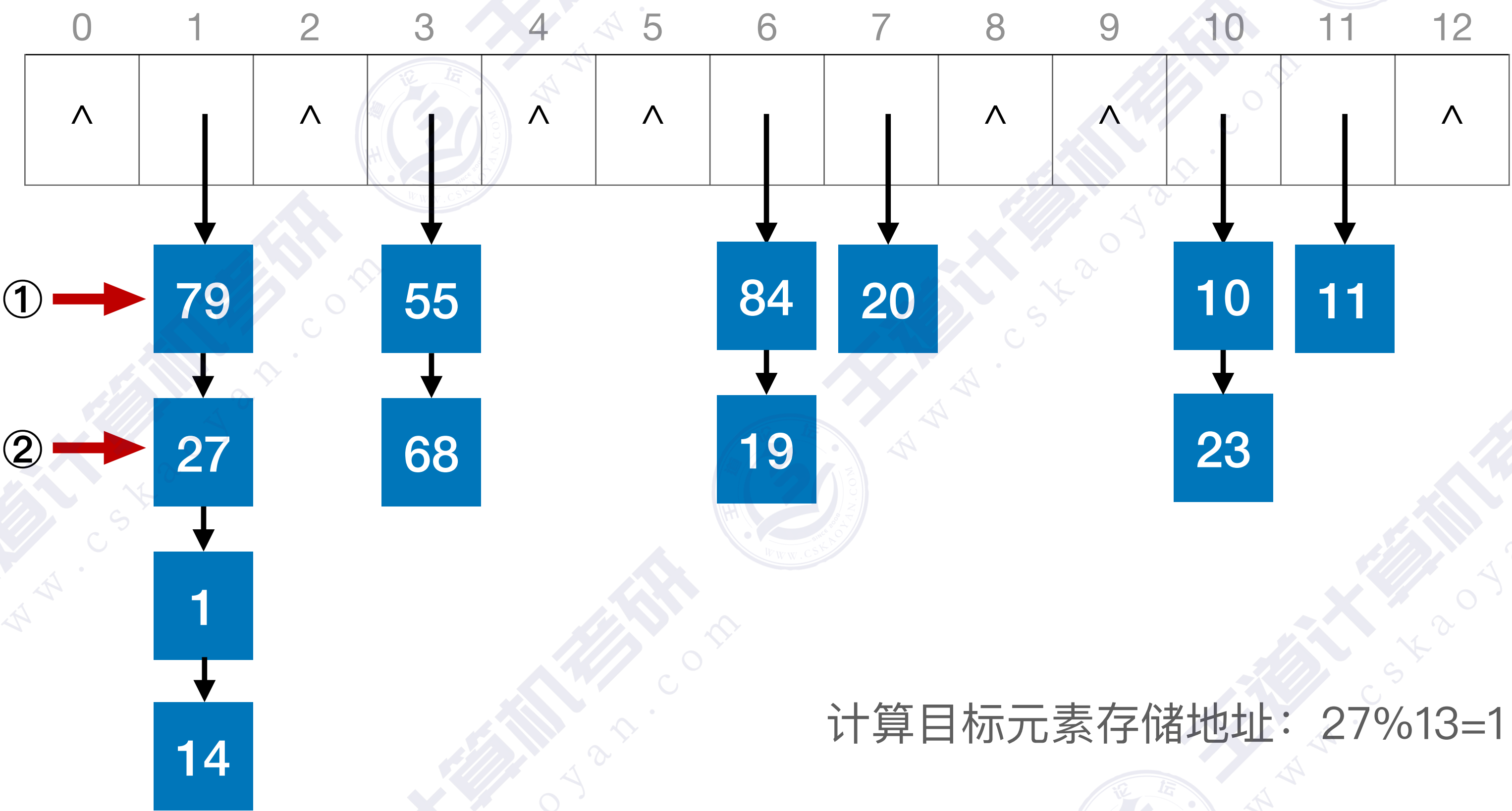
21查找失败，查找长度=0

注：有的教材会把“空指针的对比”也计入查找长度。但考试中默认只统计“关键字对比次数”。

查找长度——在查找运算中，需要对比关键字的次数称为查找长度

散列表的删除操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。在下述散列表中依次删除元素 27，20，66。

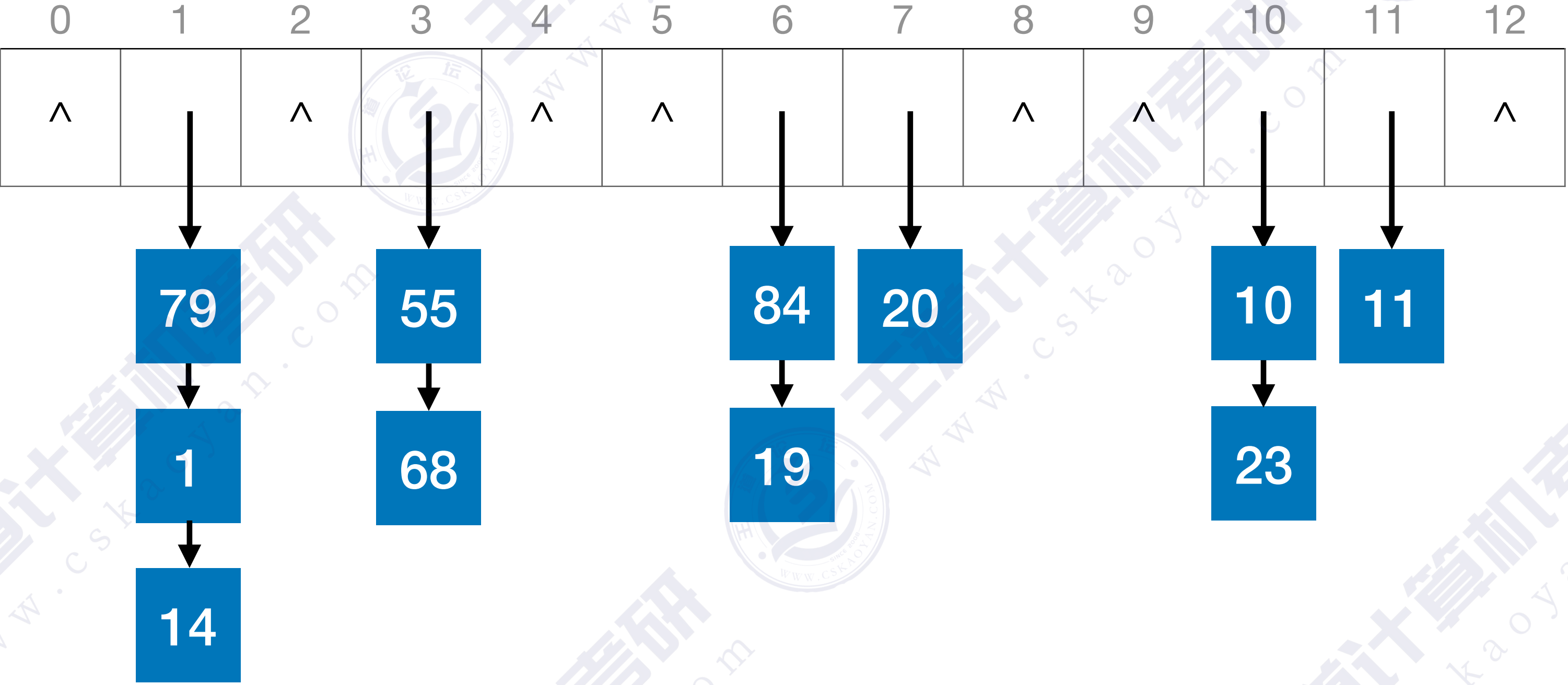


删除目标：27

计算目标元素存储地址： $27\%13=1$

散列表的删除操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。在下述散列表中依次删除元素 27，20，66。



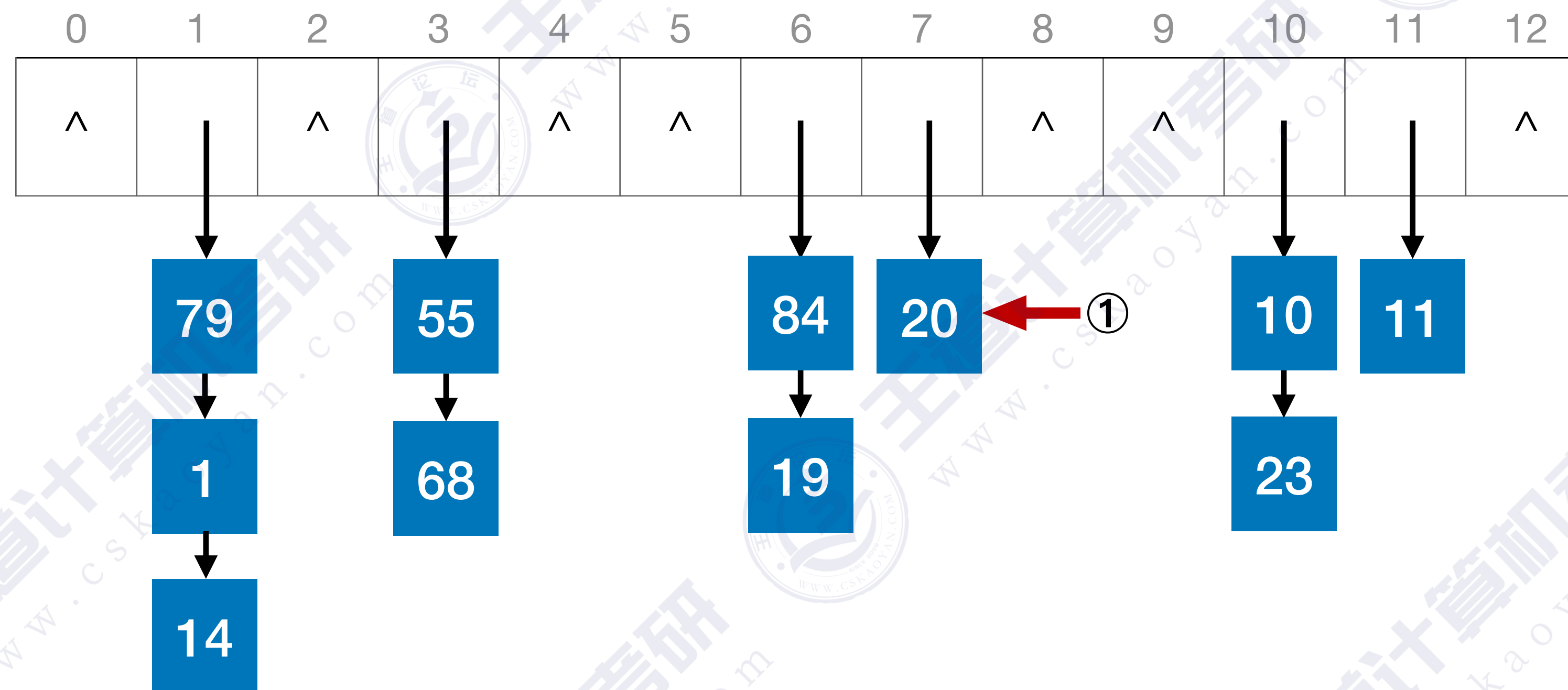
删除目标：**27**

计算目标元素存储地址： $27\%13=1$

27查找成功，删除成功

散列表的删除操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(\text{key}) = \text{key} \% 13$ ，用拉链法解决冲突。在下述散列表中依次删除元素 27, 20, 66。

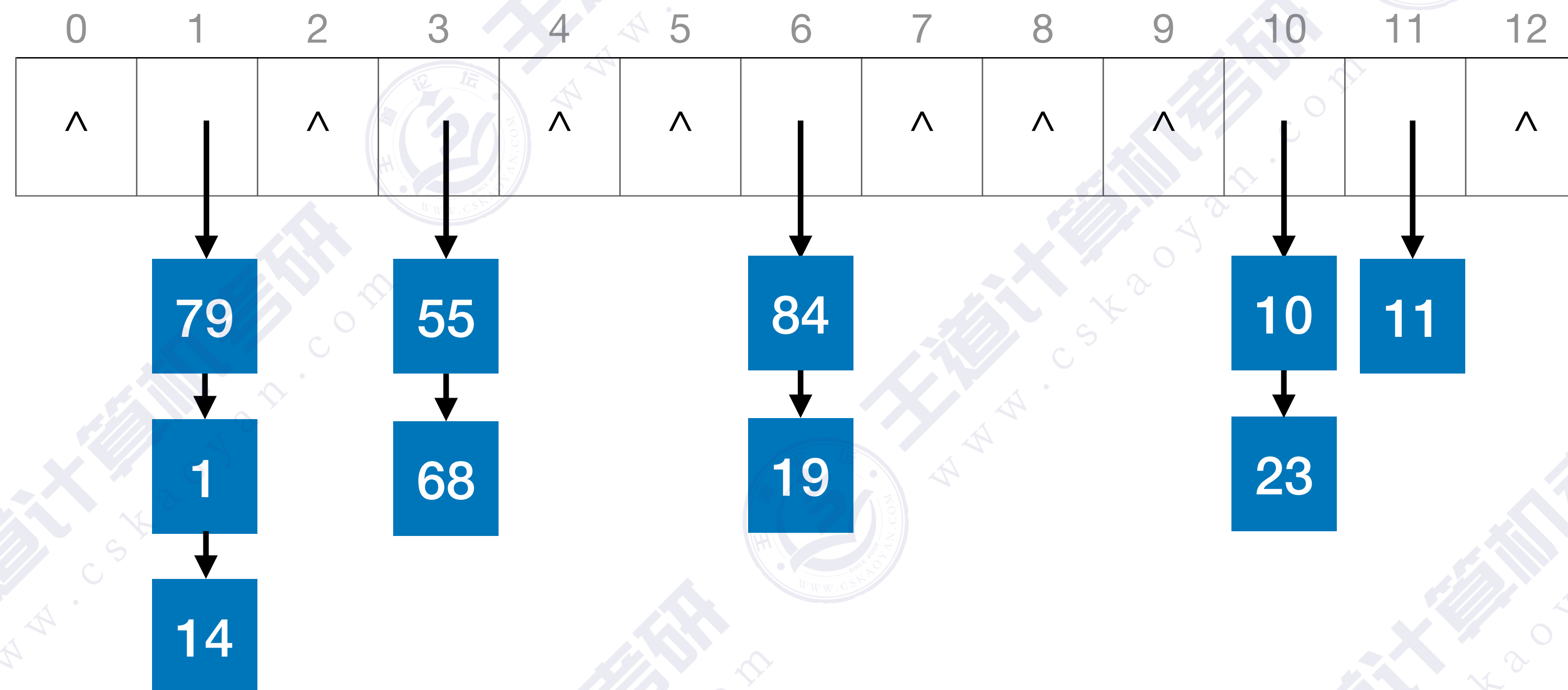


删除目标：20

计算目标元素存储地址： $20 \% 13 = 7$

散列表的删除操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。在下述散列表中依次删除元素 27，20，66。



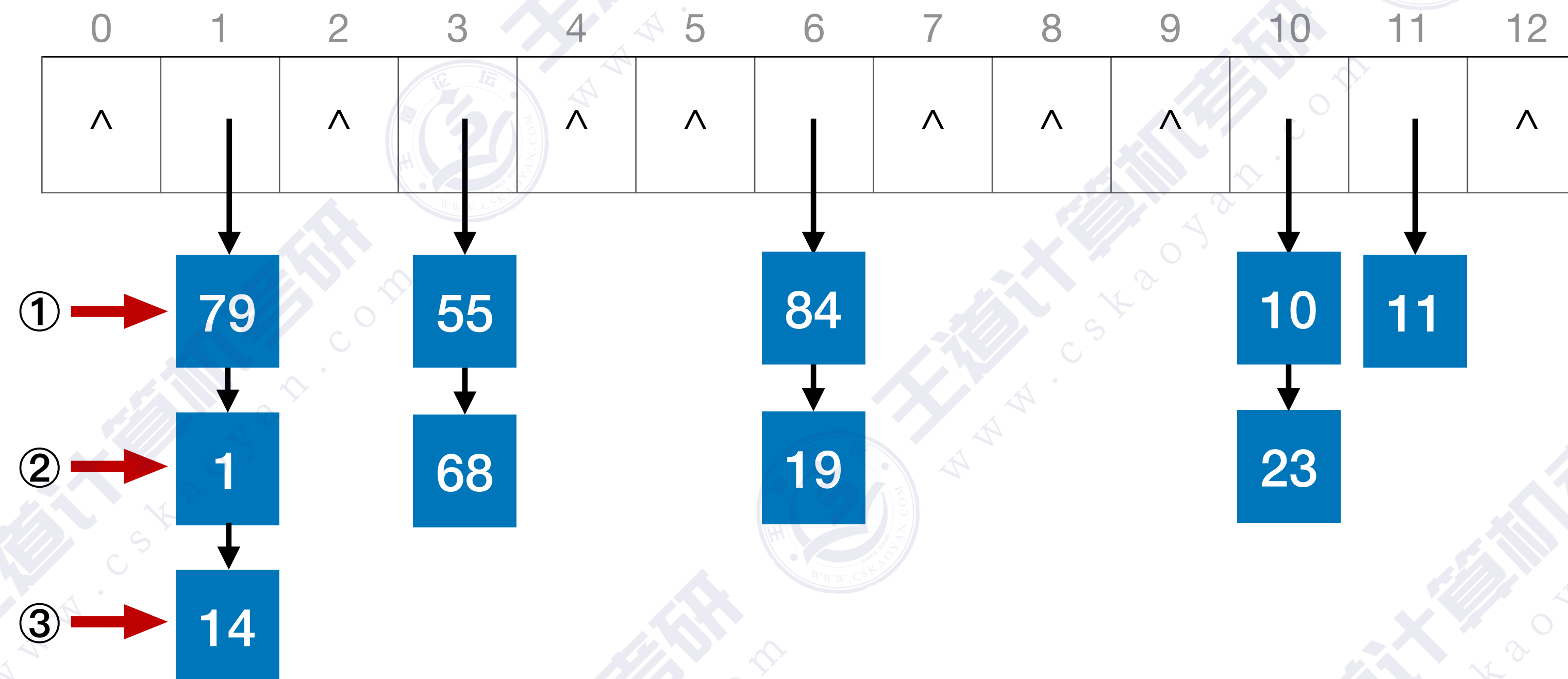
删除目标：20

计算目标元素存储地址： $20\%13=7$

20查找成功，删除成功

散列表的删除操作（拉链法解决冲突）

例：若散列表长度为13，散列函数 $H(\text{key}) = \text{key} \% 13$ ，用拉链法解决冲突。在下述散列表中依次删除元素 27, 20, 66。



删除目标：66

计算目标元素存储地址： $66 \% 13 = 1$

66查找失败，删除失败

知识回顾与重要考点

处理冲突的方法 ——拉链法

拉链法（又称链接法、链地址法）：把所有“同义词”存储在一个链表中

插入操作

Step 1：根据散列函数计算新元素的散列地址

Step 2：将新元素插入散列地址对应的链表（可用头插法或尾插法）

查找操作

Step 1：根据散列函数计算目标元素的散列地址

Step 2：顺序查找散列地址对应的链表，直到查找成功或查找失败

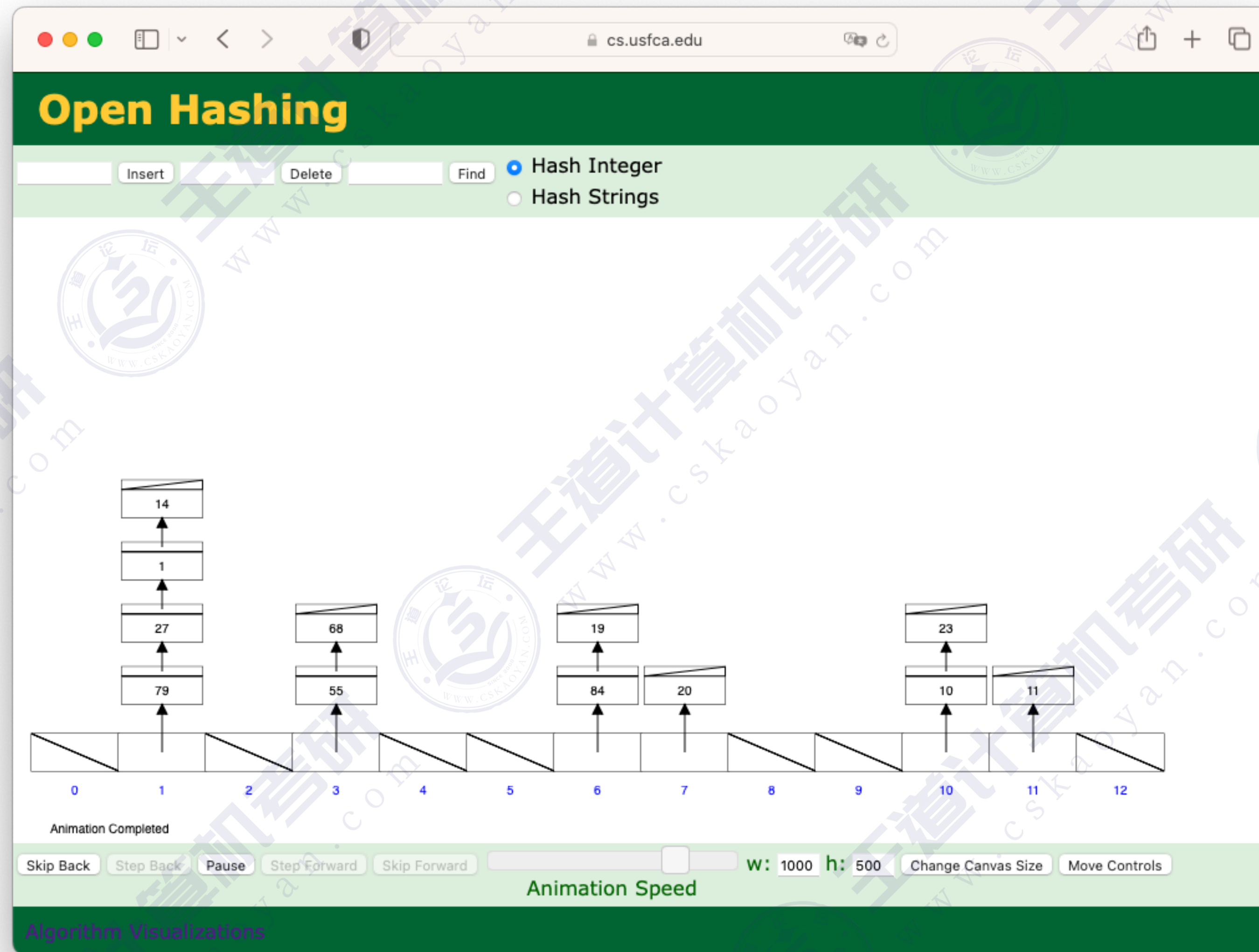
在分析查找长度时，通常只统计“关键字的对比次数”，而链表“空指针的对比次数”不计入查找长度

删除操作

Step 1：根据散列函数计算目标元素的散列地址

Step 2：顺序查找散列地址对应的链表，若查找成功，将目标元素从链表中删除

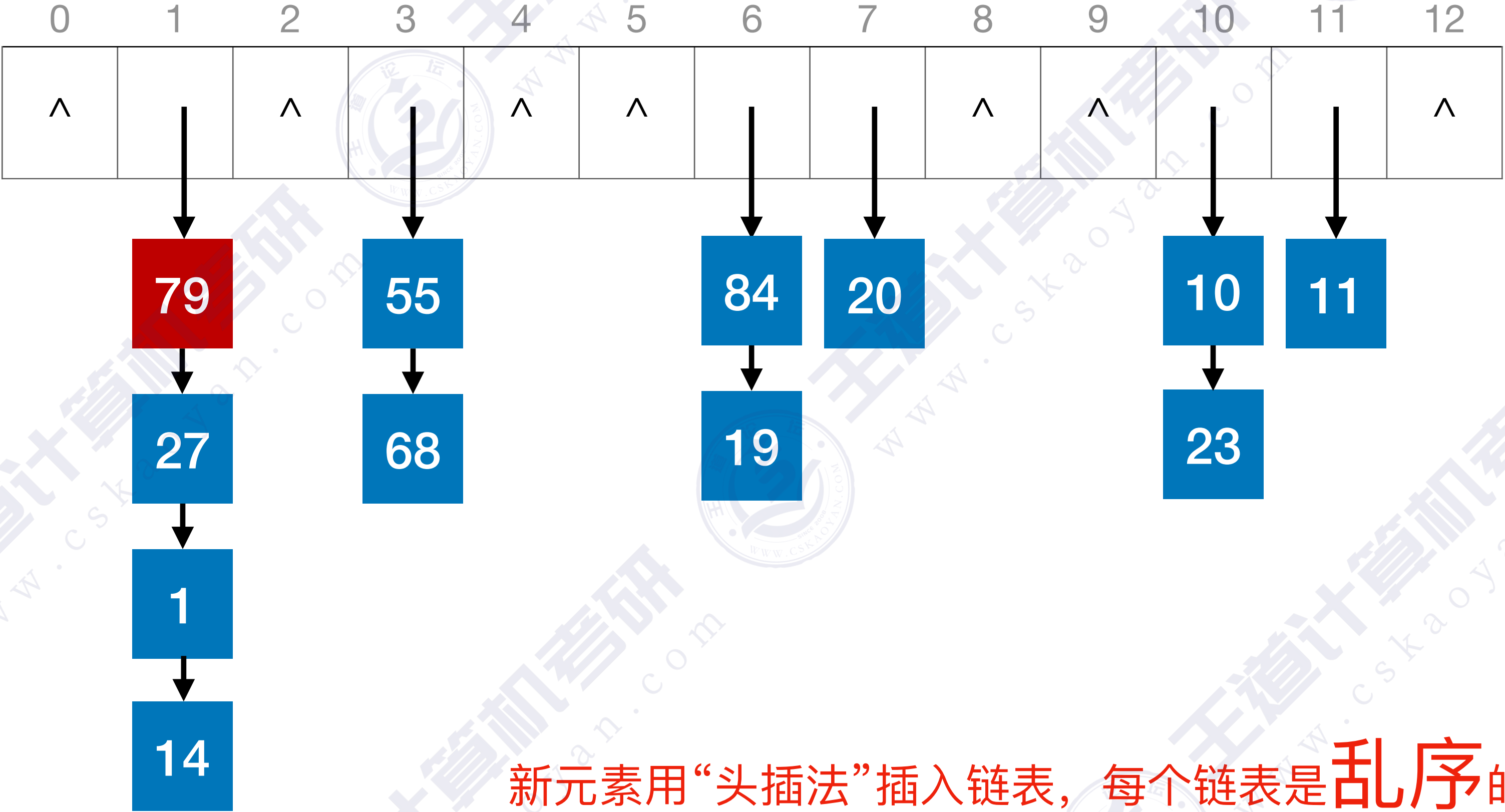
传送门



<https://www.cs.usfca.edu/~galles/visualization/OpenHash.html>

拓展：插入操作的小优化

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。依次插入关键字 {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}



19%13=6

14%13=1

23%13=10

1%13=1

68%13=3

20%13=7

84%13=6

27%13=1

55%13=3

11%13=11

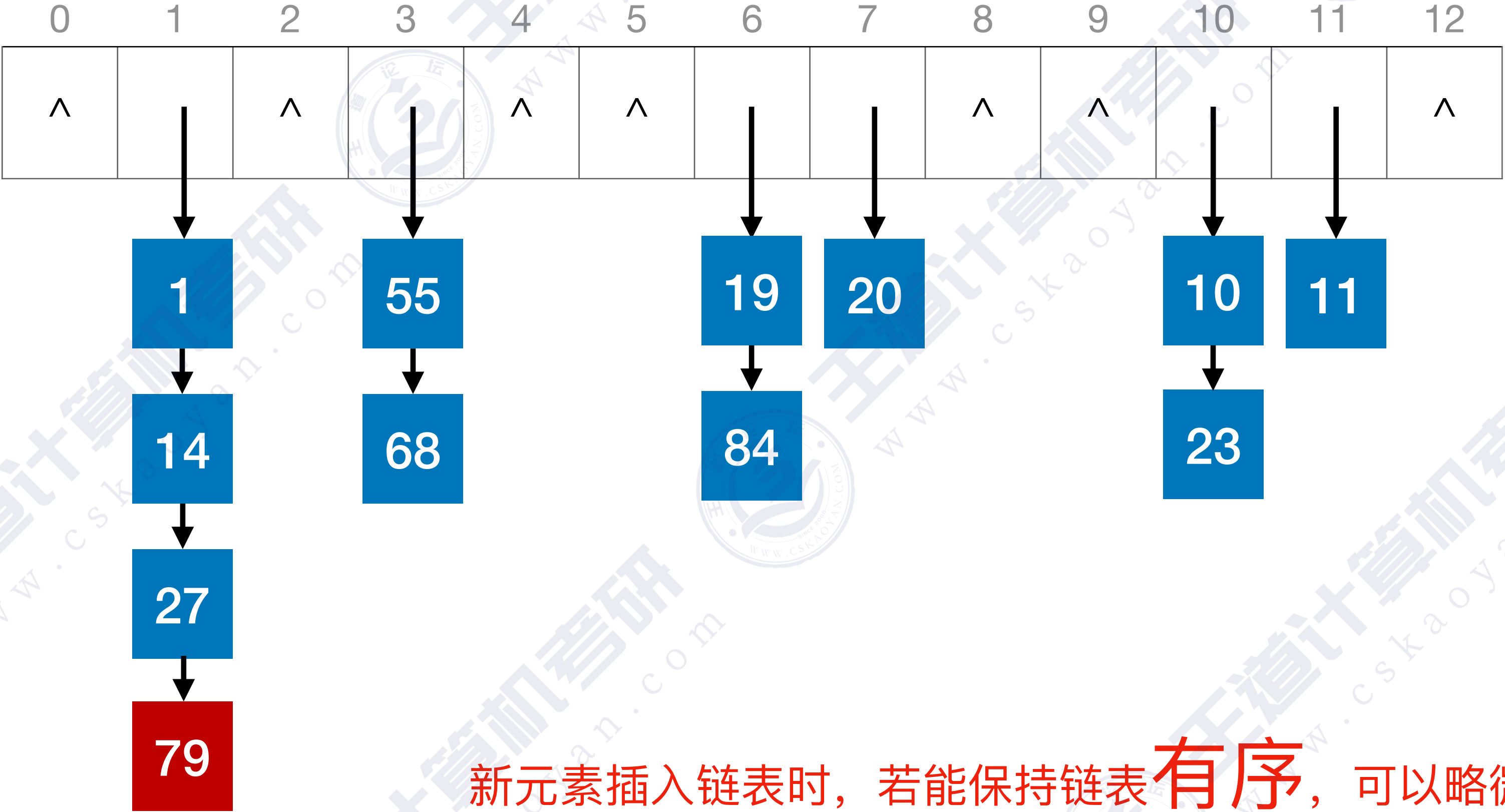
10%13=10

79%13=1

新元素用“头插法”插入链表，每个链表是乱序的。

拓展：插入操作的小优化

例：若散列表长度为13，散列函数 $H(key)=key\%13$ ，用拉链法解决冲突。依次插入关键字 {19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}



$19\%13=6$	$27\%13=1$
$14\%13=1$	$55\%13=3$
$23\%13=10$	$11\%13=11$
$1\%13=1$	$10\%13=10$
$68\%13=3$	$79\%13=1$
$20\%13=7$	
$84\%13=6$	

新元素插入链表时，若能保持链表**有序**，可以略微提高“查找”效率。