

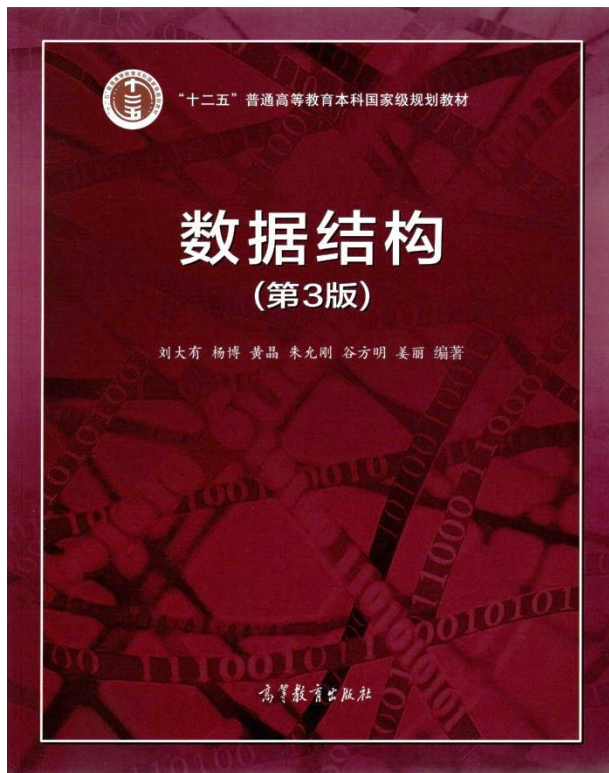


计算机学院王湘浩班  
2024级



# 分布排序

- 桶排序
- 基数排序



数据之法  
结构之美  
算法之道

Last updated on 2025.6

zhuyungang@jlu.edu.cn



邓明扬

麻省理工学院21级本科生

2020年NOI全国中学生信息学奥赛决赛金牌

2021年IOI世界中学生信息学奥赛冠军

2022年ICPC国际大学生程序设计竞赛全球总决赛冠军

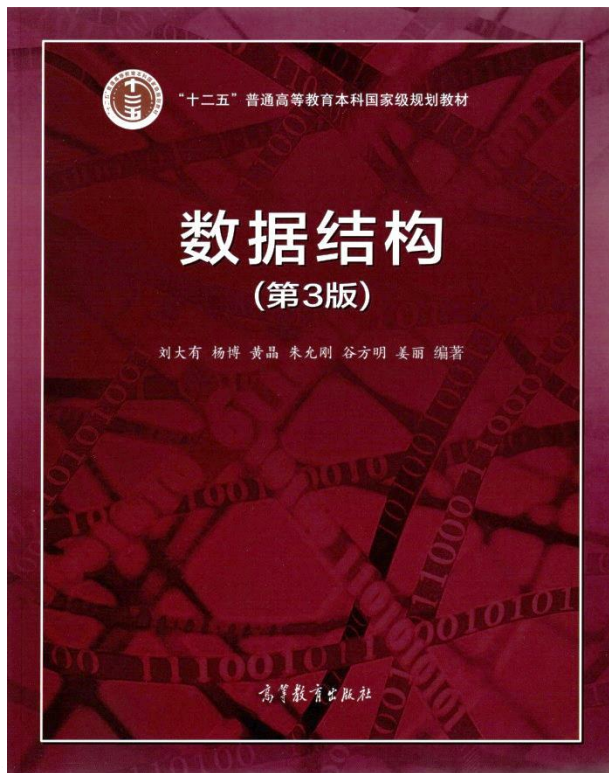
如果你是一个初学者的话，  
要尽量自己调代码，我知道很多  
小伙伴都喜欢找别人帮你调代码，  
但这个东西最后是需要自己练的。  
要多练习，不仅锻炼打代码  
的速度和准确性，也锻炼思考问  
题的能力。





# 分布排序

- 桶排序
- 基数排序

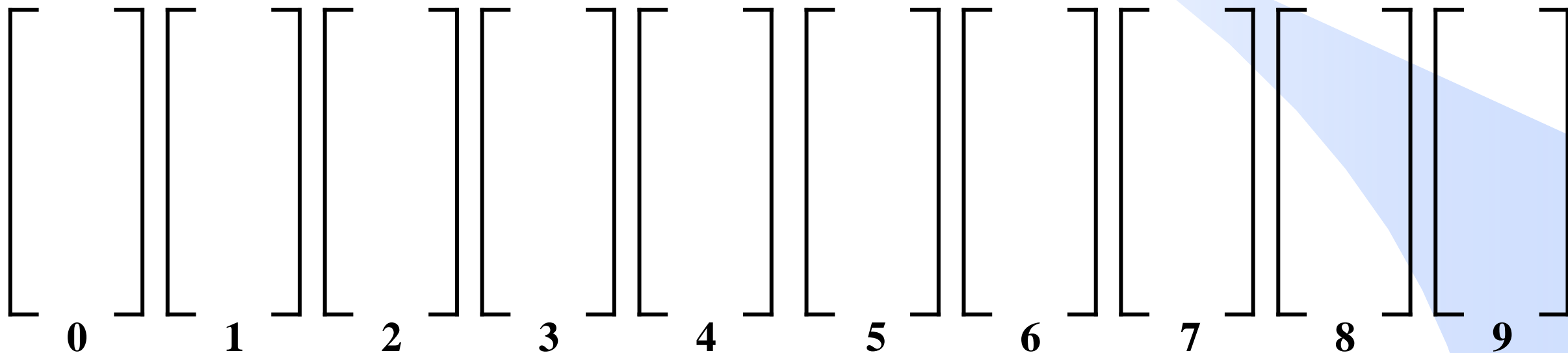


数据之法  
结构之美  
算法之道

# 桶排序 (Bucket Sort)

待排序的文件 $R$ 包含 $n$ 个整数，每个整数的值域为 $[0, m)$

9 6 3 5 2 0 1 5\*

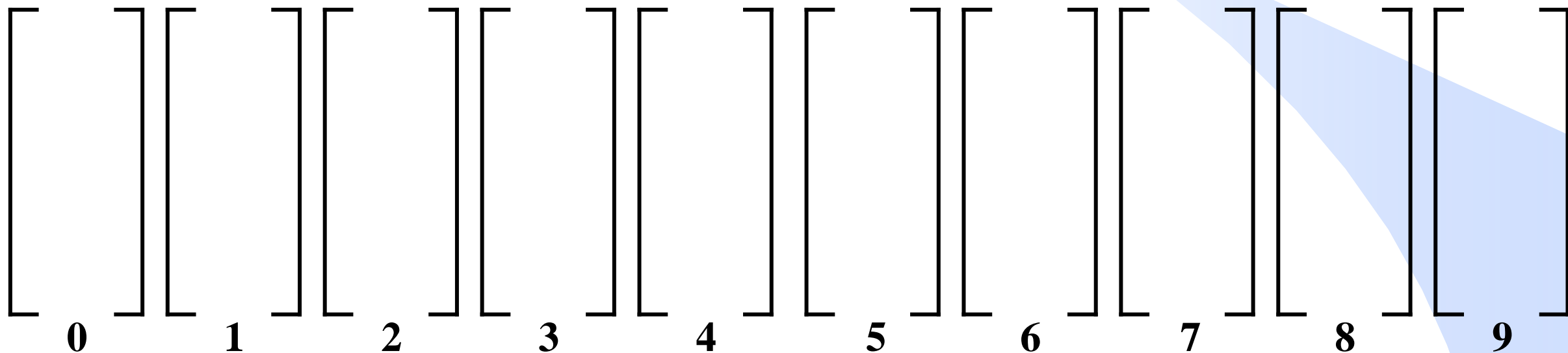


设置  $m$  个桶，对应元素的值域  $0 \dots m-1$

# 桶排序

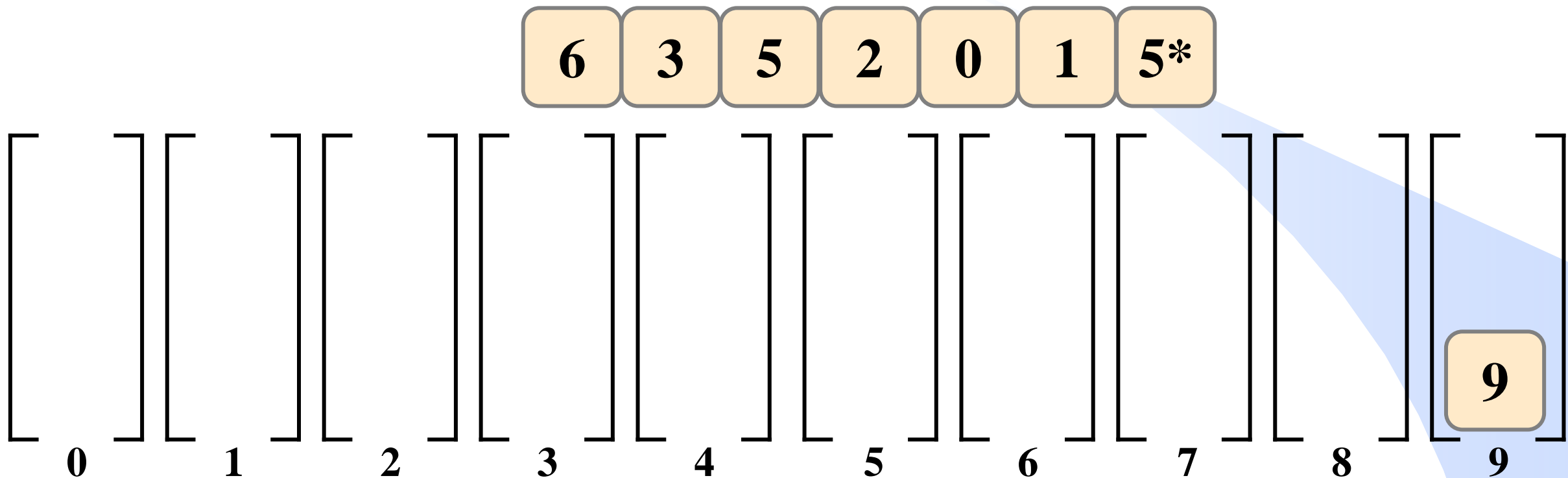
① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中

9 6 3 5 2 0 1 5\*



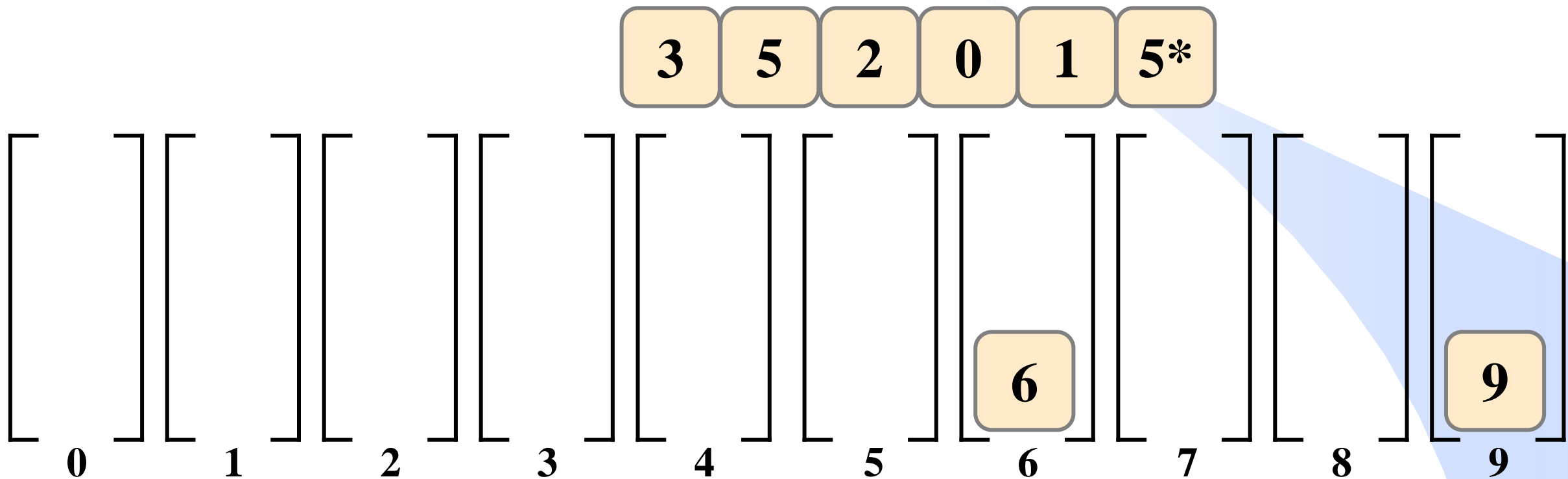
# 桶排序

① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中



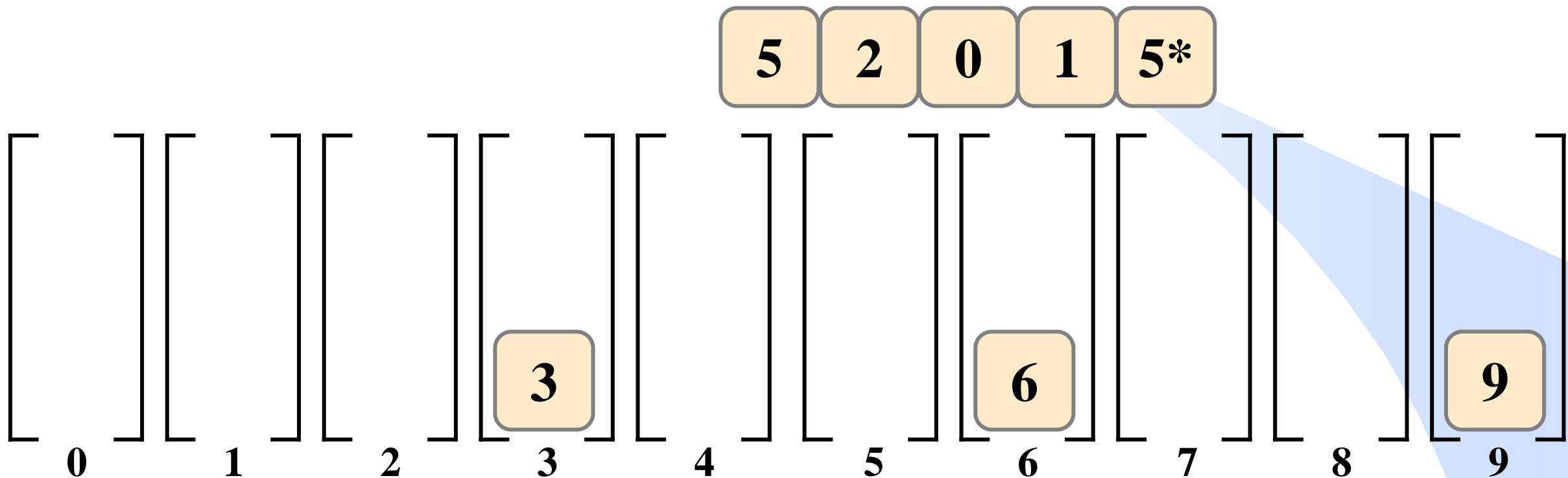
# 桶排序

① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中



# 桶排序

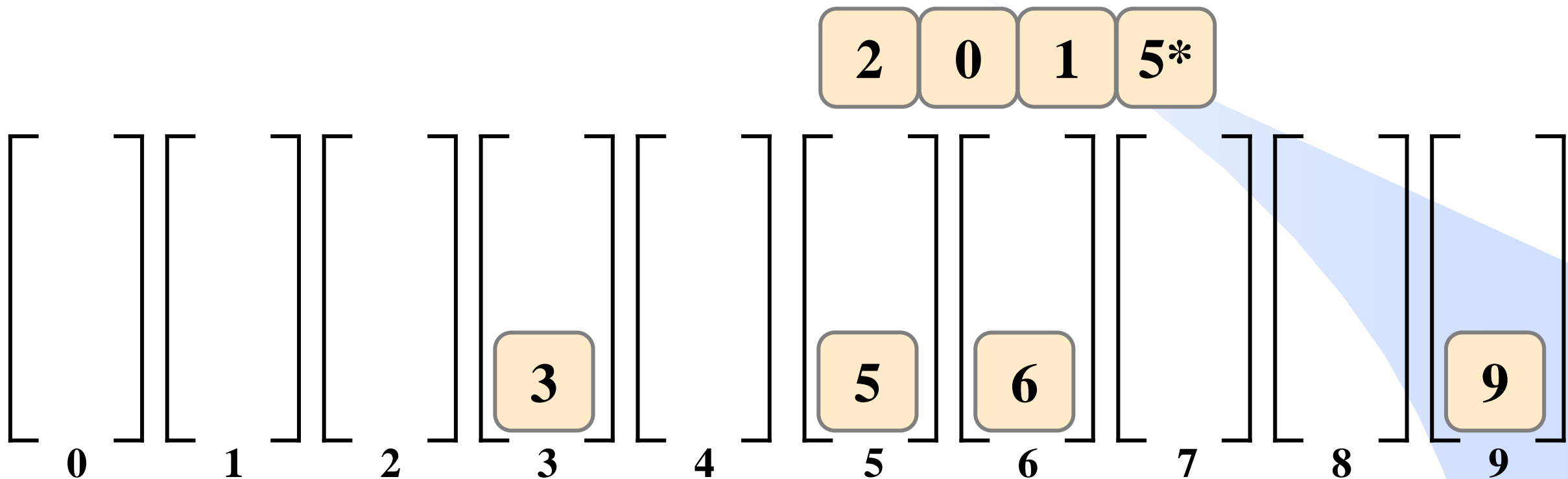
① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中





# 桶排序

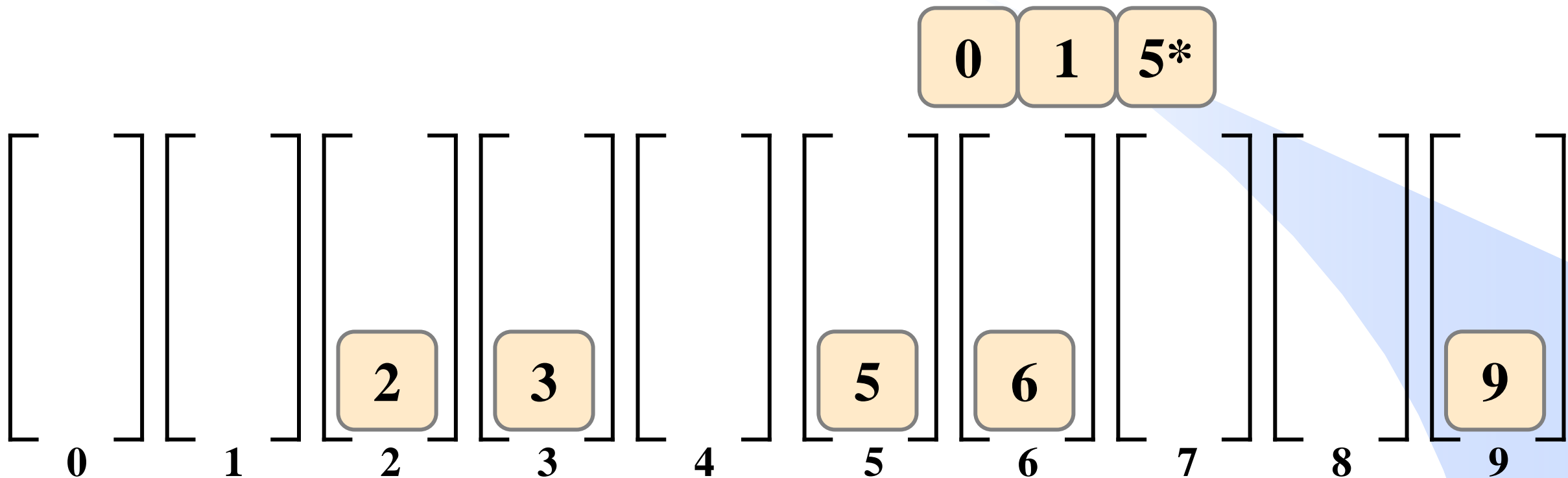
① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中



# 桶排序

A

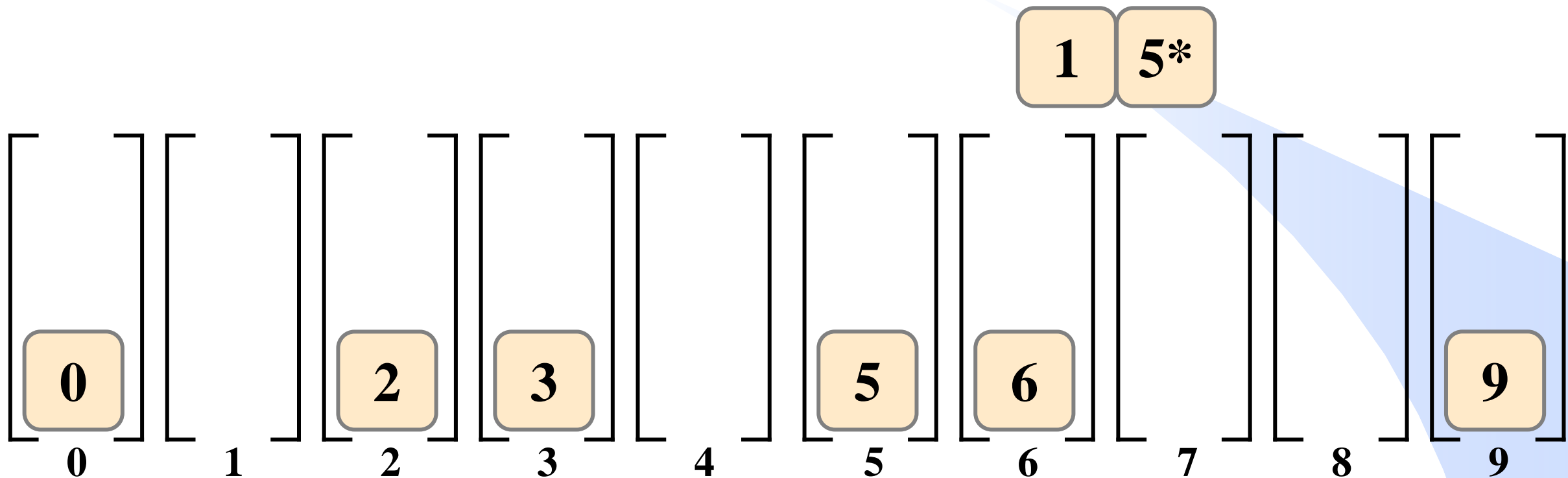
① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中



# 桶排序

A

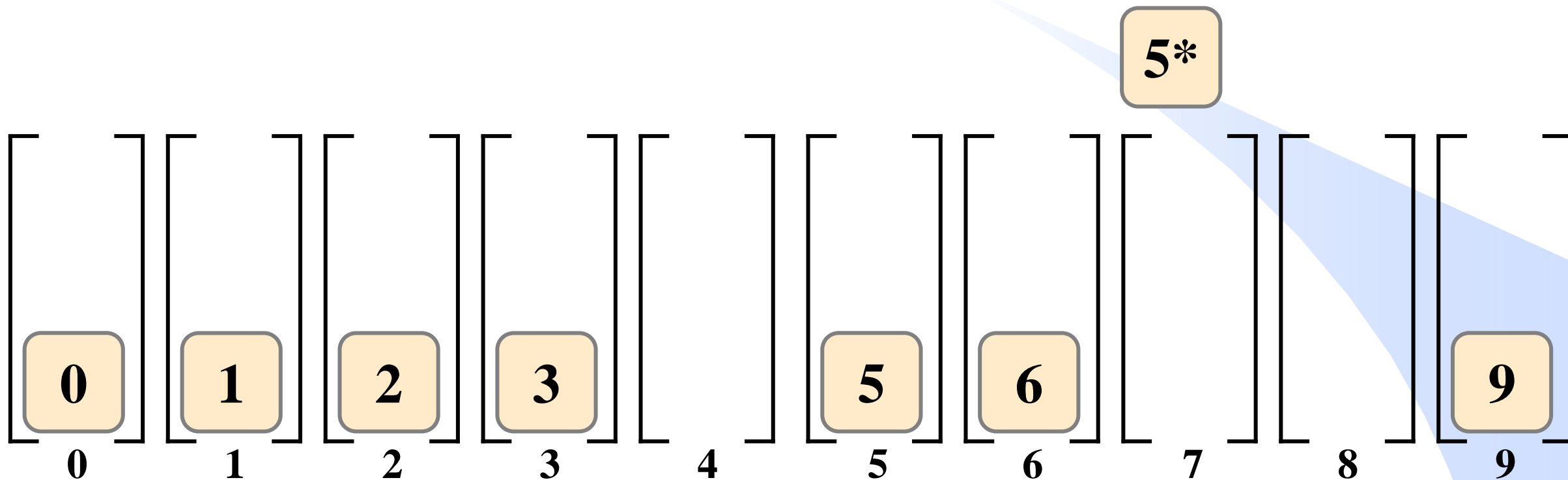
① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中



# 桶排序

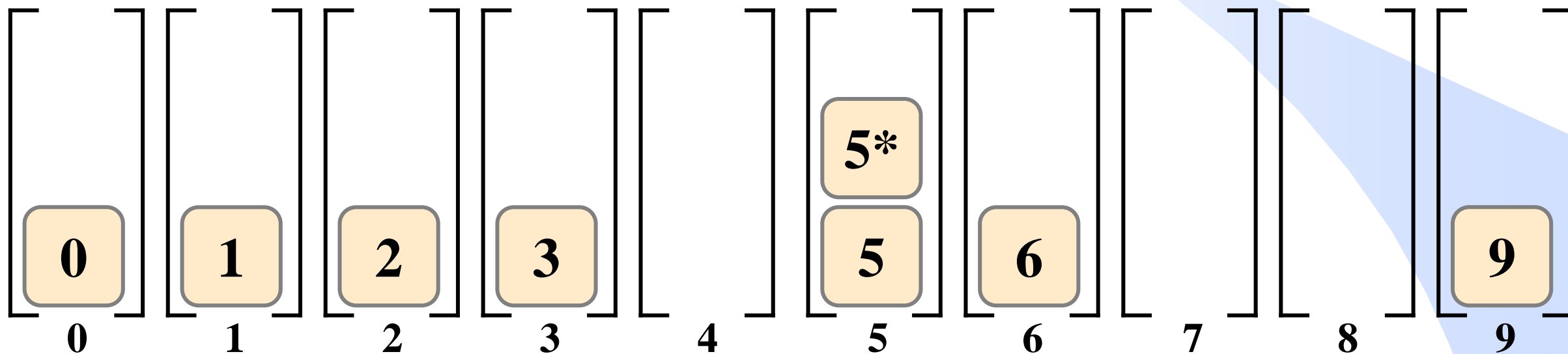
A

① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中



# 桶排序

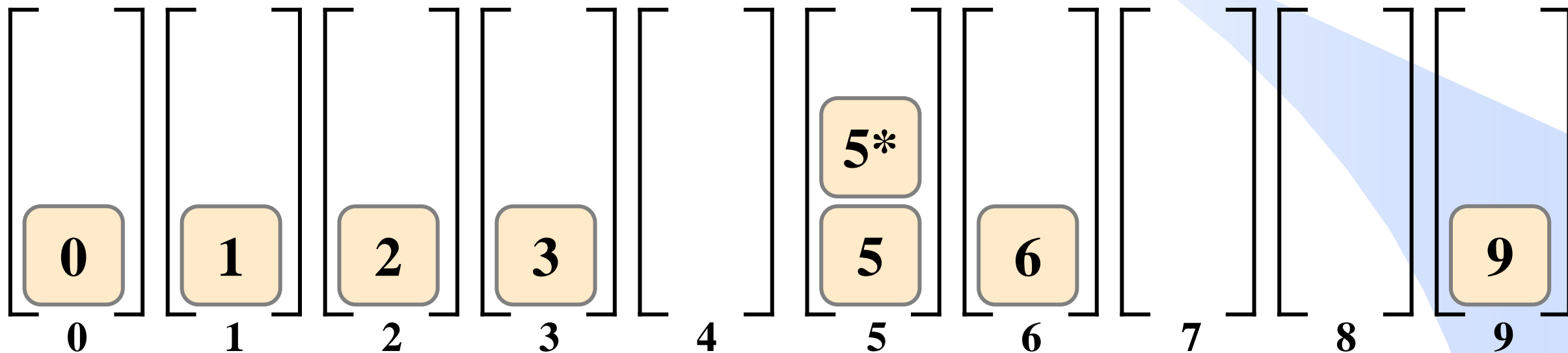
① 分配：扫描 $n$ 个元素，按关键词放入对应的桶中



# 桶排序

A

② 收集：依次将桶中元素取出

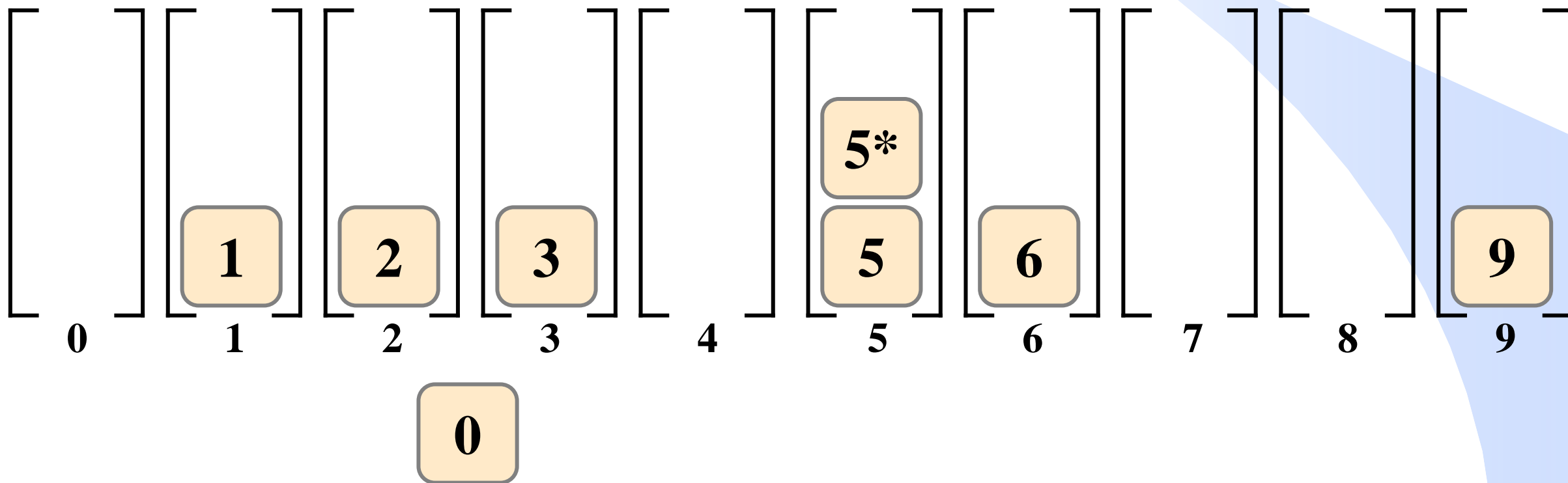




# 桶排序

A

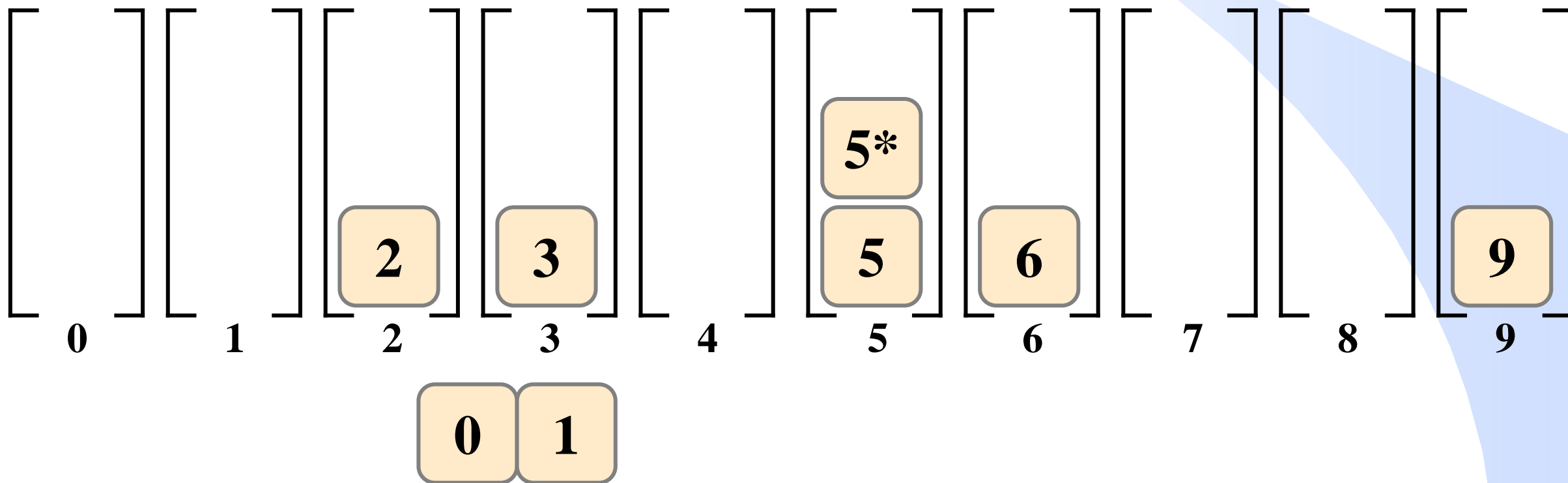
② 收集：依次将桶中元素取出



# 桶排序

A

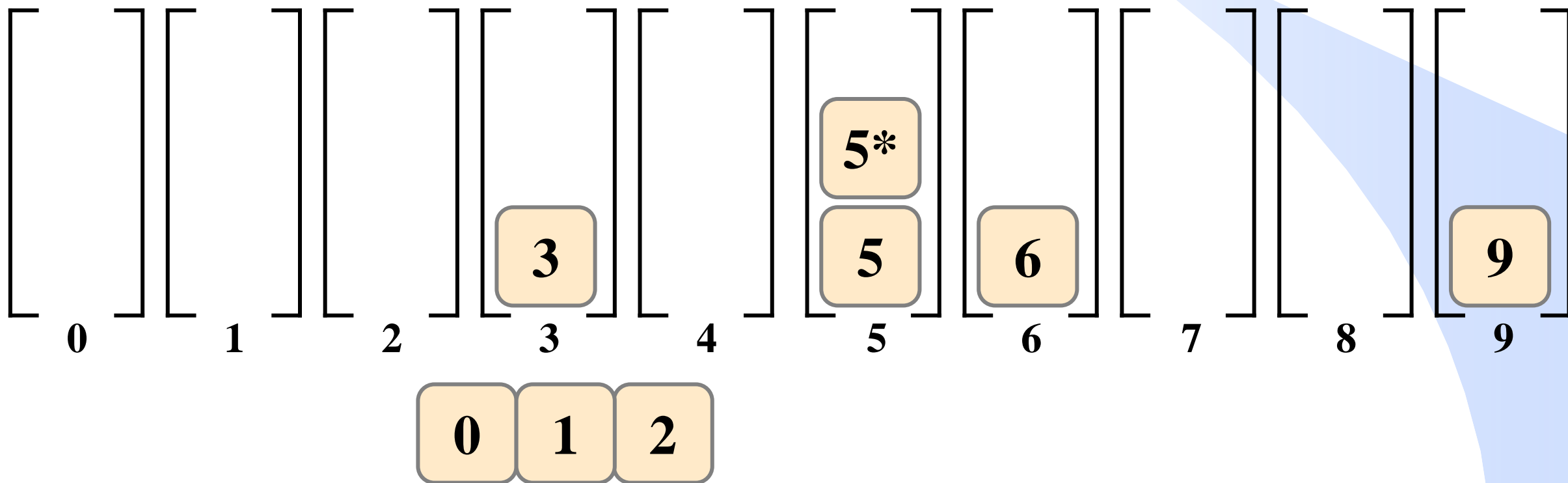
② 收集：依次将桶中元素取出



# 桶排序

A

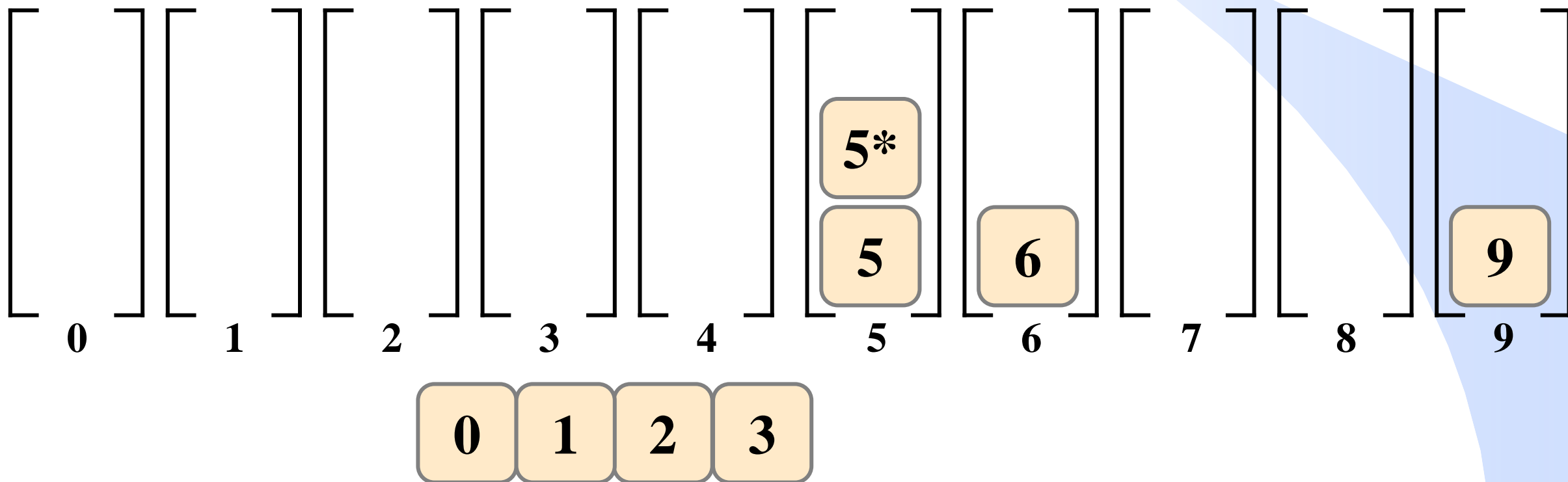
② 收集：依次将桶中元素取出



# 桶排序

A

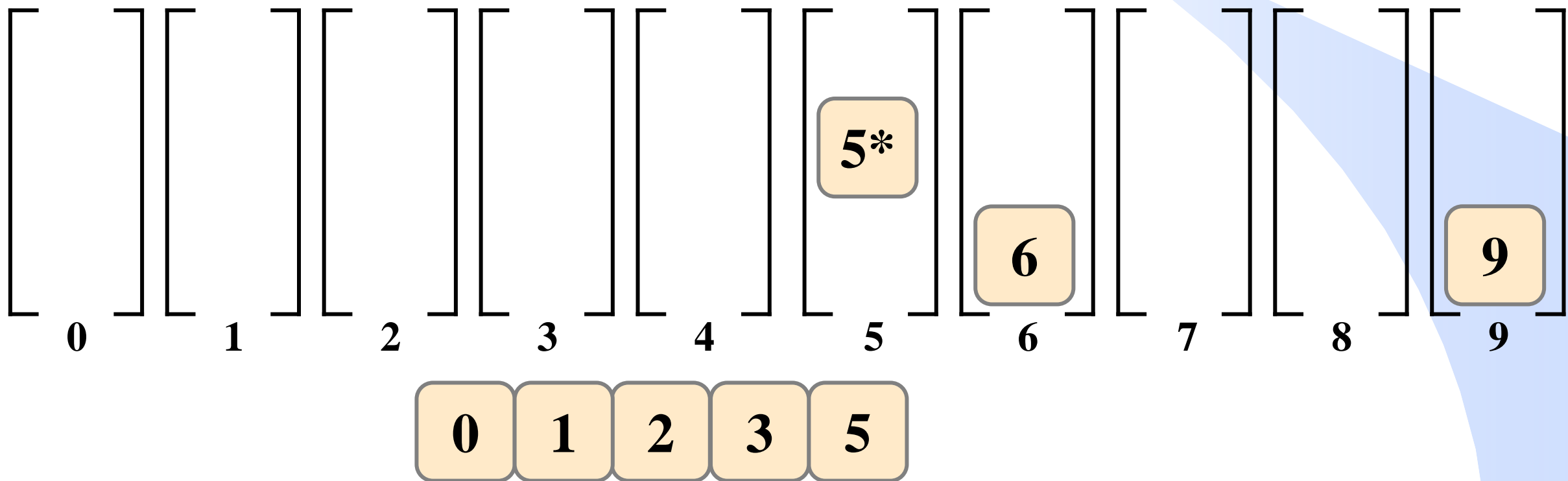
② 收集：依次将桶中元素取出



# 桶排序

A

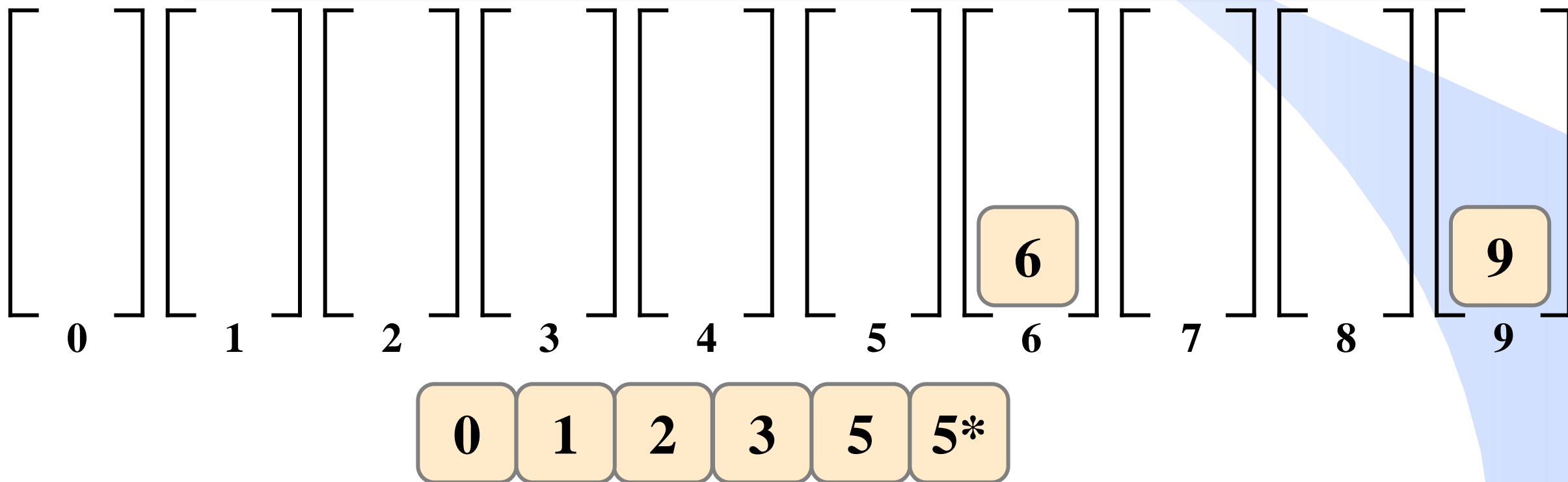
② 收集：依次将桶中元素取出



# 桶排序

## ② 收集：依次将桶中元素取出

桶排序是稳定的：关键词相同的元素在同一桶里，排在前面的元素在分配时先入桶，收集时也先出桶

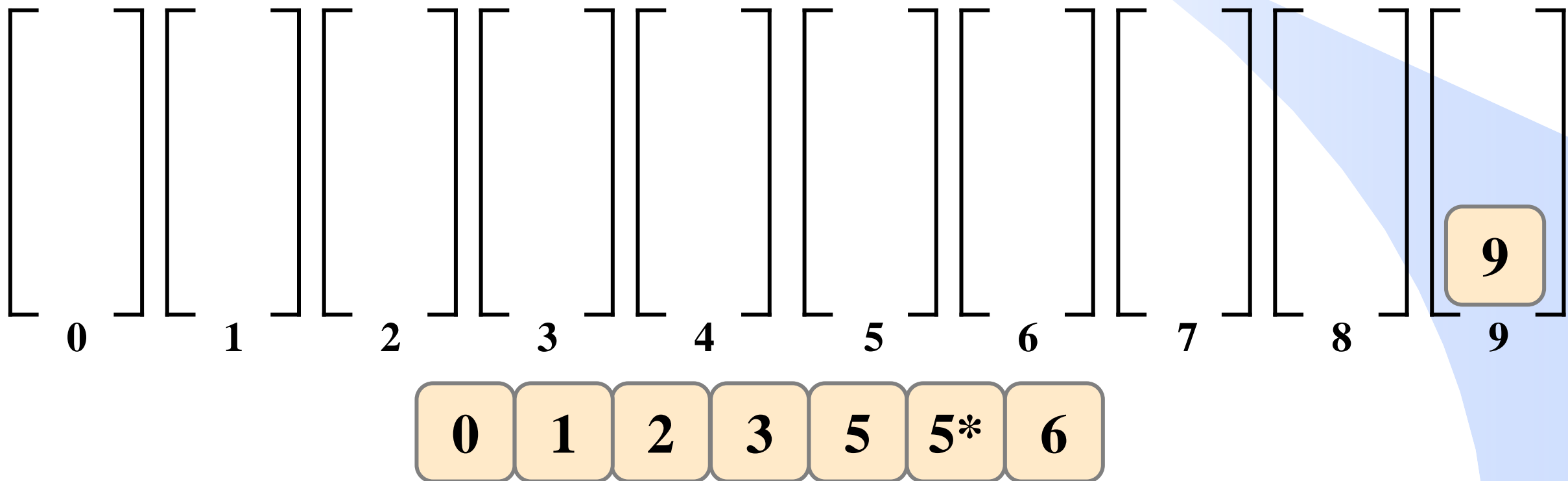




# 桶排序

A

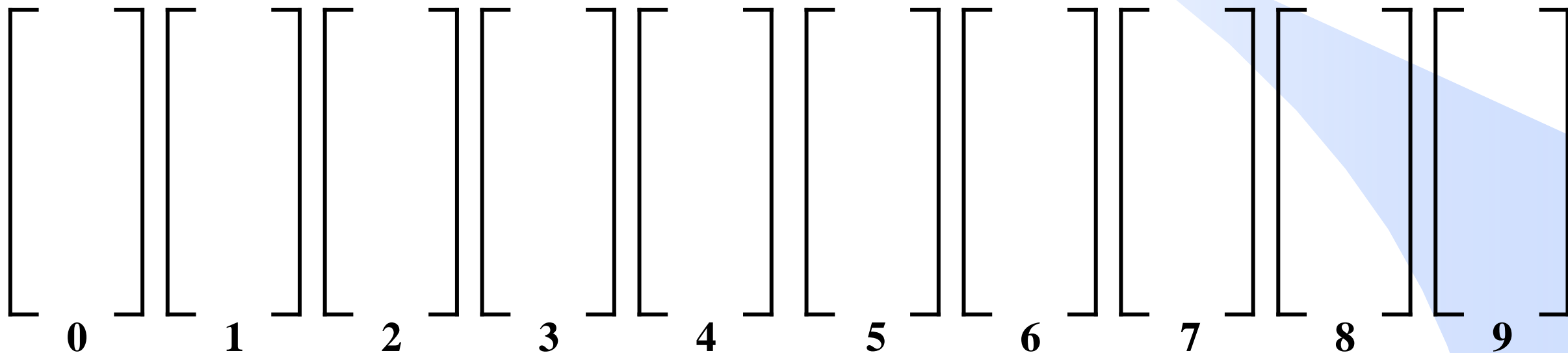
② 收集：依次将桶中元素取出



# 桶排序

- ① 分配：将 $n$ 个元素依次放入桶中
- ② 收集：依次将 $m$ 个桶中的元素取出

时空复杂度  
 $O(n+m)$



桶的实现  
链式队列



稳定

# 桶排序总结

排序算法	时间复杂度			空间复杂度	稳定性
	最好	平均	最坏		
桶排序	$O(n+m)$	$O(n+m)$	$O(n+m)$	$O(n+m)$	稳定

每个元素的值域为 $[0, m)$ , 即 $m$ 为桶的个数

若 $m=O(n)$ , 则时空复杂度为线性

## 思考

**B**

- 对吉林大学24级全体同学按英语考试分数排序，哪种排序算法更好？
  - A. 快速排序
  - B. 桶排序
- 对世界各国的人口进行排序，哪种排序算法更好？
  - A. 快速排序
  - B. 桶排序

## 思考

➤ 对10个整数排序，每个整数的范围为 $[0, 1000)$ 。

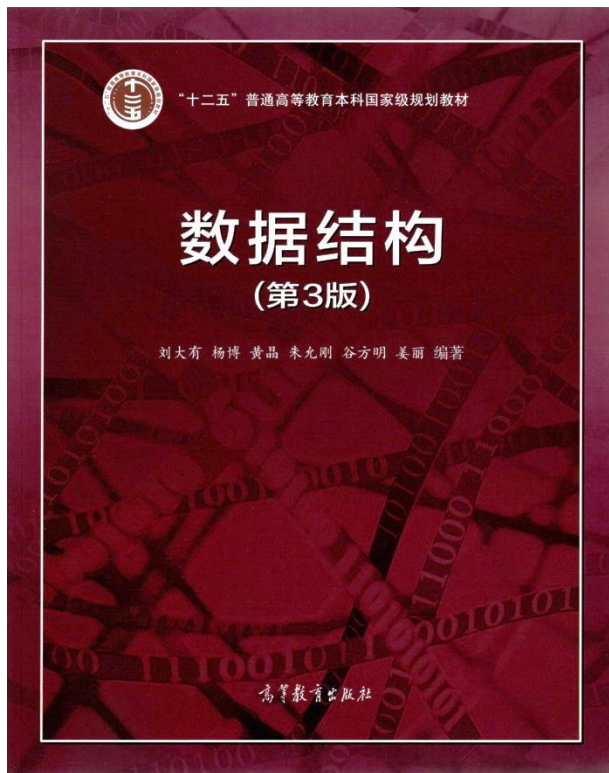
29 257 658 839 236 720 56 237 999 155

$n=10$   
 $m=1000$



# 分布排序

- 桶排序
- 基数排序



数据之法  
结构之美  
算法之道



# 基数排序



**Herman Hollerith**  
**(1860-1929)**  
**发明打孔卡制表机**  
**IBM公司之父**

# 基数排序 (Radix Sort)

- 元素的关键词由多个域构成，即  $K = K_d, K_{d-1}, \dots, K_2, K_1$ 
  - ✓ 若每个域为英文字母，则关键词即英文单词
  - ✓ 若每个域为1位十进制数字(0~9)，则关键词即  $d$  位十进制数
- 自  $K_1$  至  $K_d$  (自低位向高位)，依次以各域为序进行稳定排序

2	9	
2	5	7
6	5	8
8	3	9
2	3	6
7	2	0
1	5	5

按个位  
(最低位)  
稳定排序



7	2	0
1	5	5
2	3	6
2	5	7
6	5	8
	2	9
8	3	9

按十位  
(次低位)  
稳定排序



十位相等的  
元素个位小  
的靠前

7	2	0
	2	9
2	3	6
8	3	9
1	5	5
2	5	7
6	5	8

按百位  
(最高位)  
稳定排序



百位相等的  
元素十位小  
的靠前

	2	9
1	5	5
2	3	6
2	5	7
6	5	8
7	2	0
8	3	9

# 基数排序

➤ 对每一位采用哪种排序方法？

✓ 要求：高效且稳定

✓ 特点：每位关键词都是整数，且在 $[0, r)$ 范围内

✓ 方法：桶排序

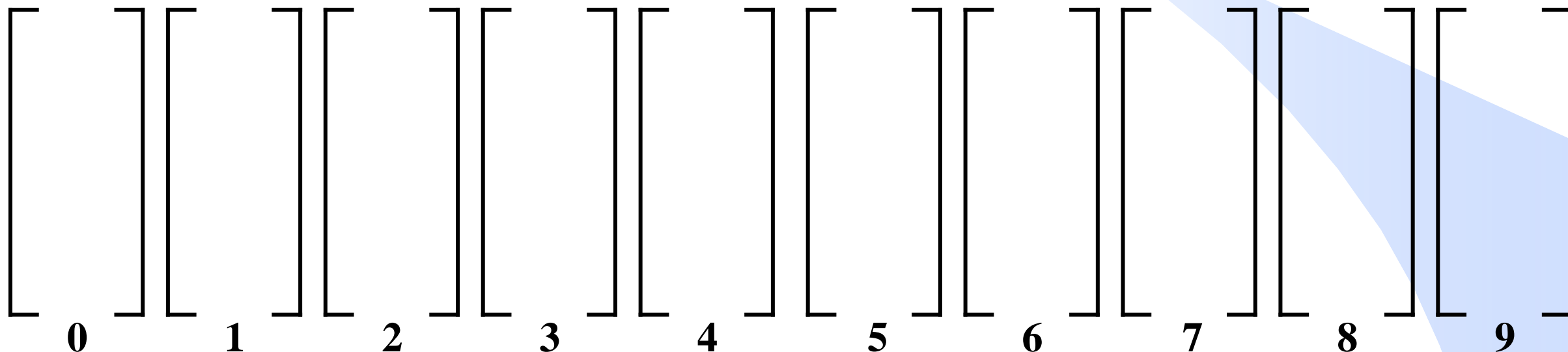
2	9	
2	5	7
6	5	8
8	3	9
2	3	6
7	2	0
1	5	5

# 基于桶排序的基数排序示例：① 按个位排序

A

29 257 658 839 236 720 155

按个位  
分配

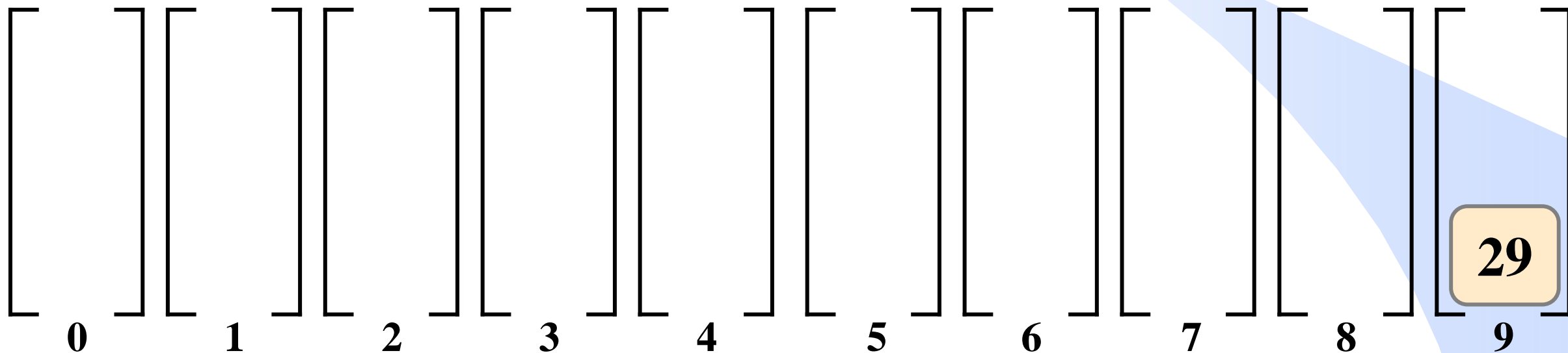


# 基于桶排序的基数排序示例：① 按个位排序

A

257 658 839 236 720 155

按个位  
分配

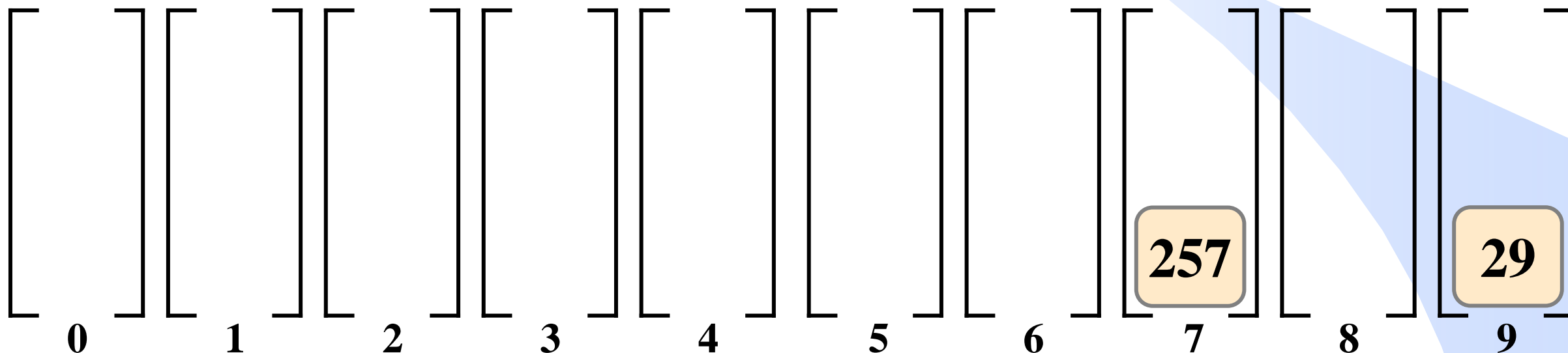


# 基于桶排序的基数排序示例：① 按个位排序

A

658 839 236 720 155

按个位  
分配





# 基于桶排序的基数排序示例：① 按个位排序

A

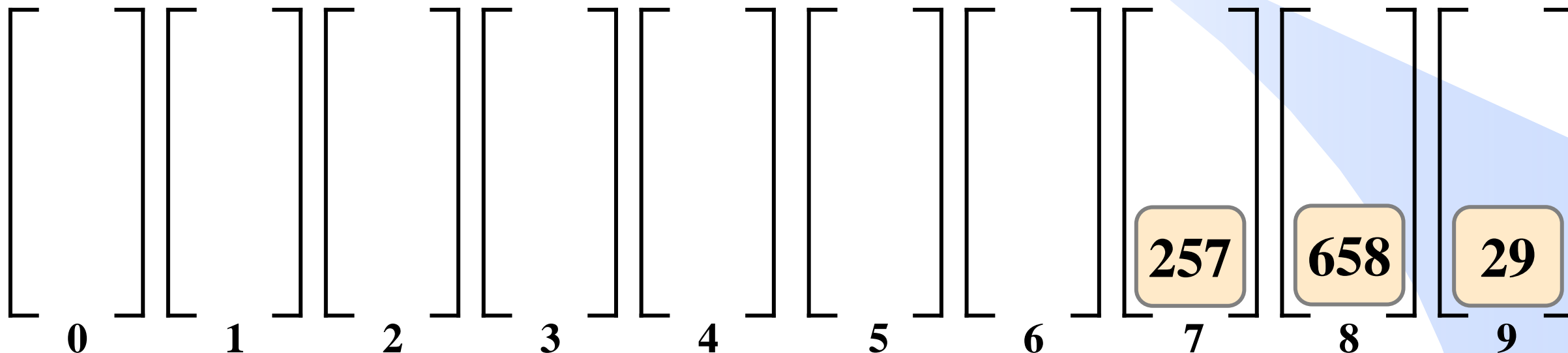
839

236

720

155

按个位  
分配

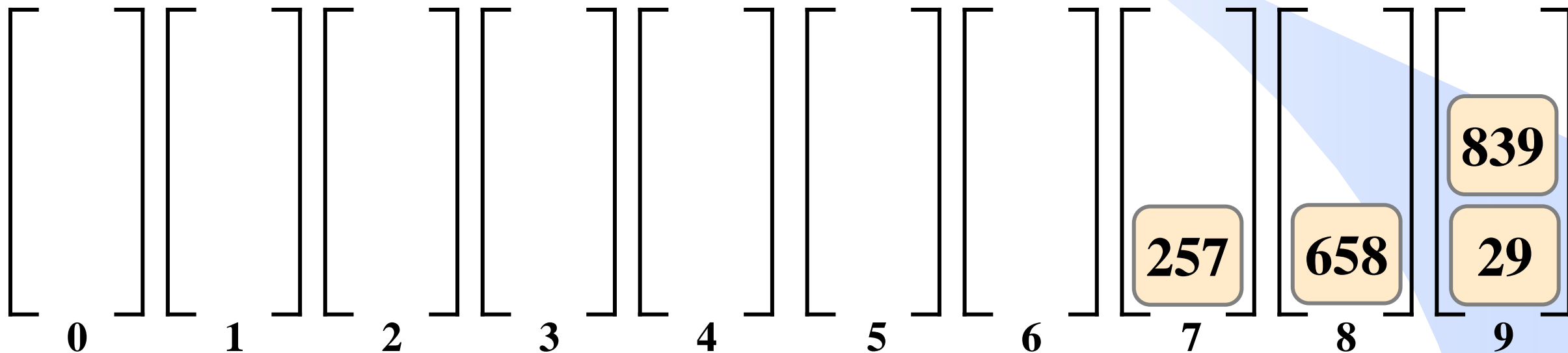


# 基于桶排序的基数排序示例：① 按个位排序

A

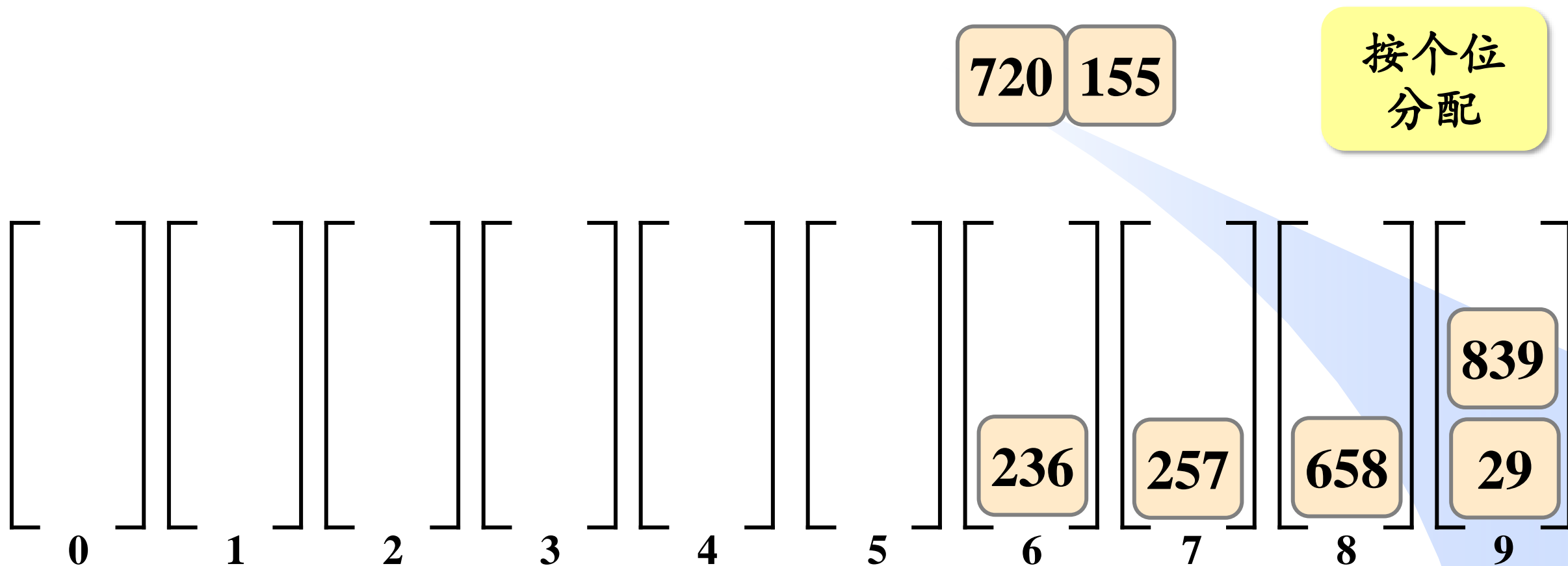
236 720 155

按个位  
分配



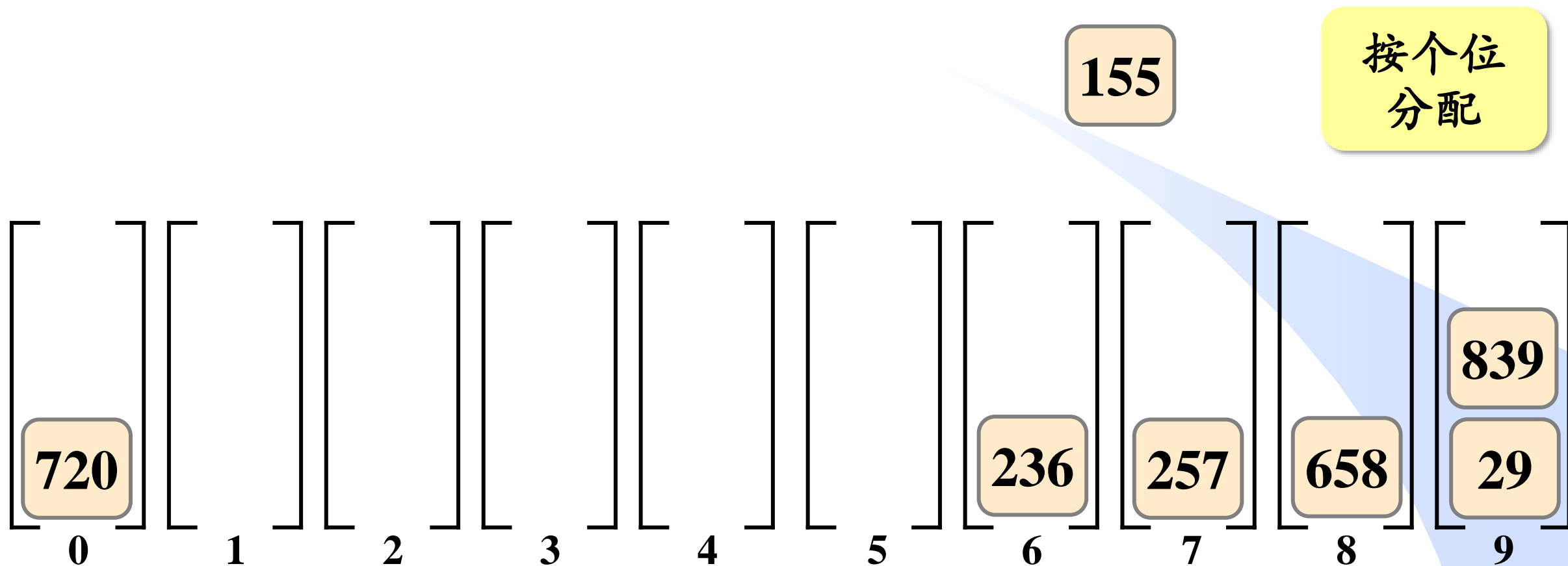
# 基于桶排序的基数排序示例：① 按个位排序

A



# 基于桶排序的基数排序示例：① 按个位排序

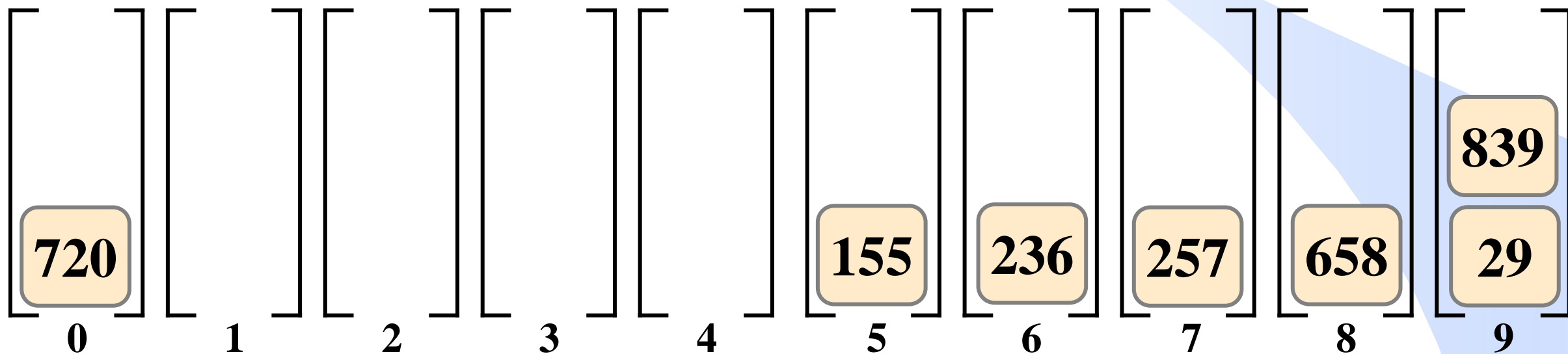
A



# 基于桶排序的基数排序示例：① 按个位排序

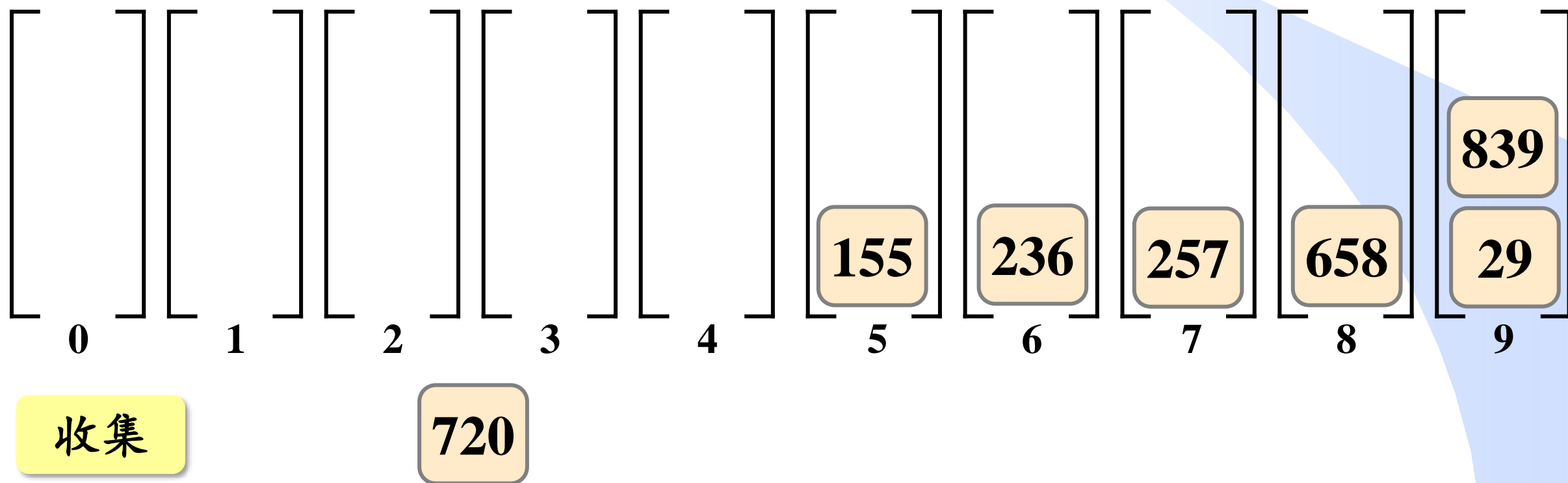
A

按个位  
分配



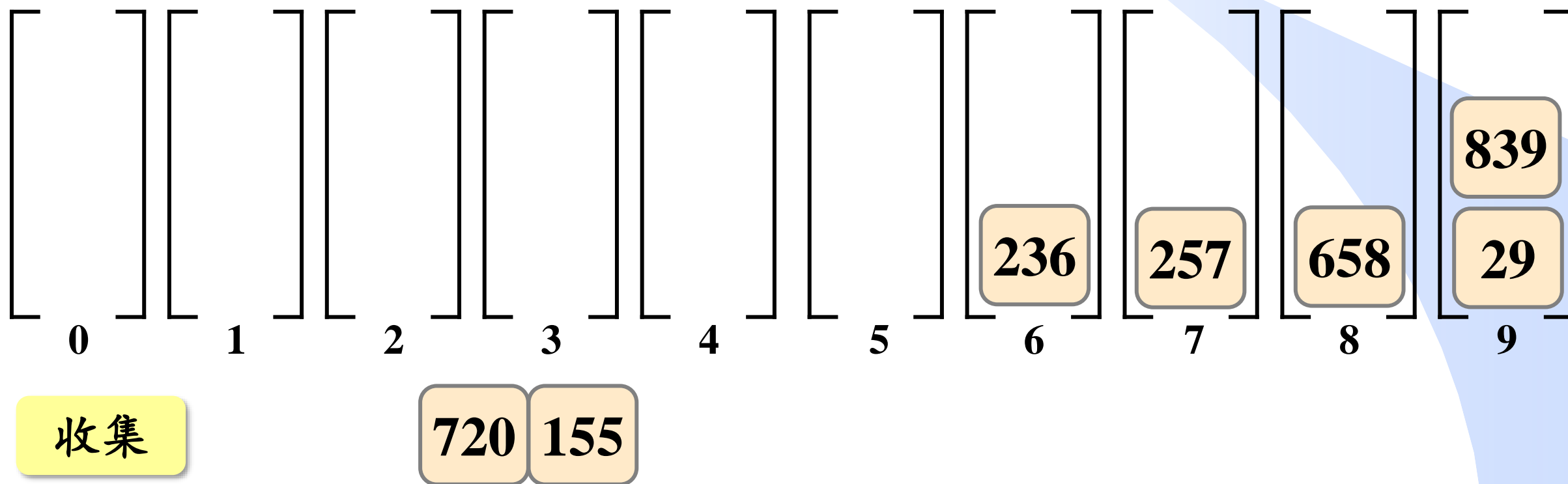
# 基于桶排序的基数排序示例：① 按个位排序

A



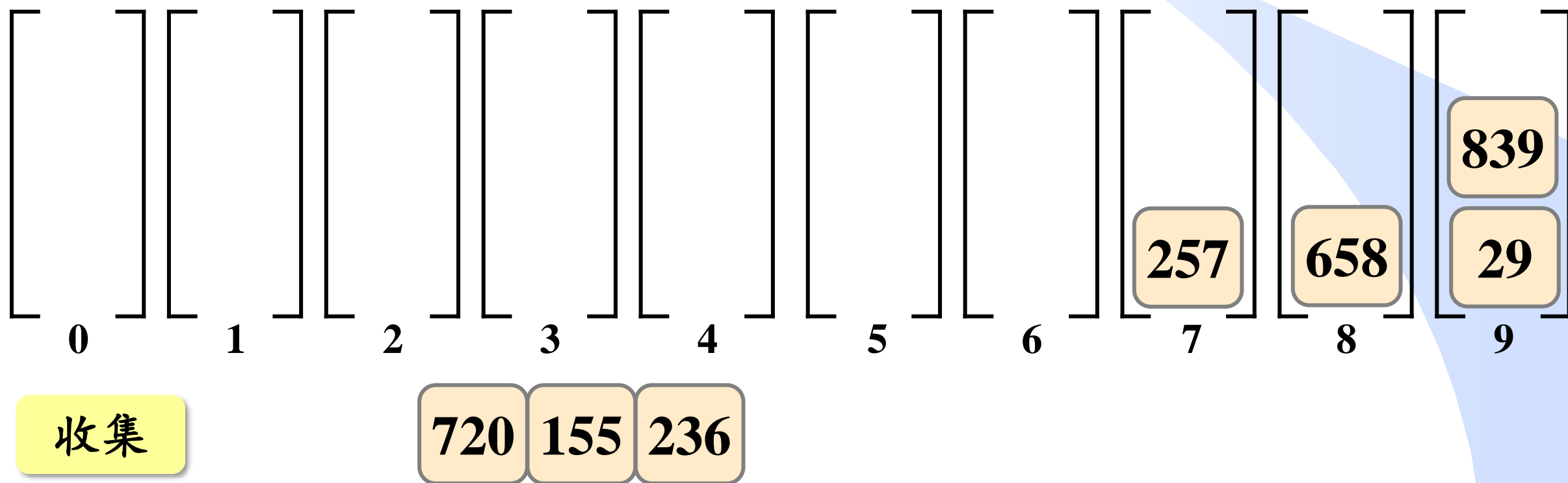
# 基于桶排序的基数排序示例：① 按个位排序

A



# 基于桶排序的基数排序示例：① 按个位排序

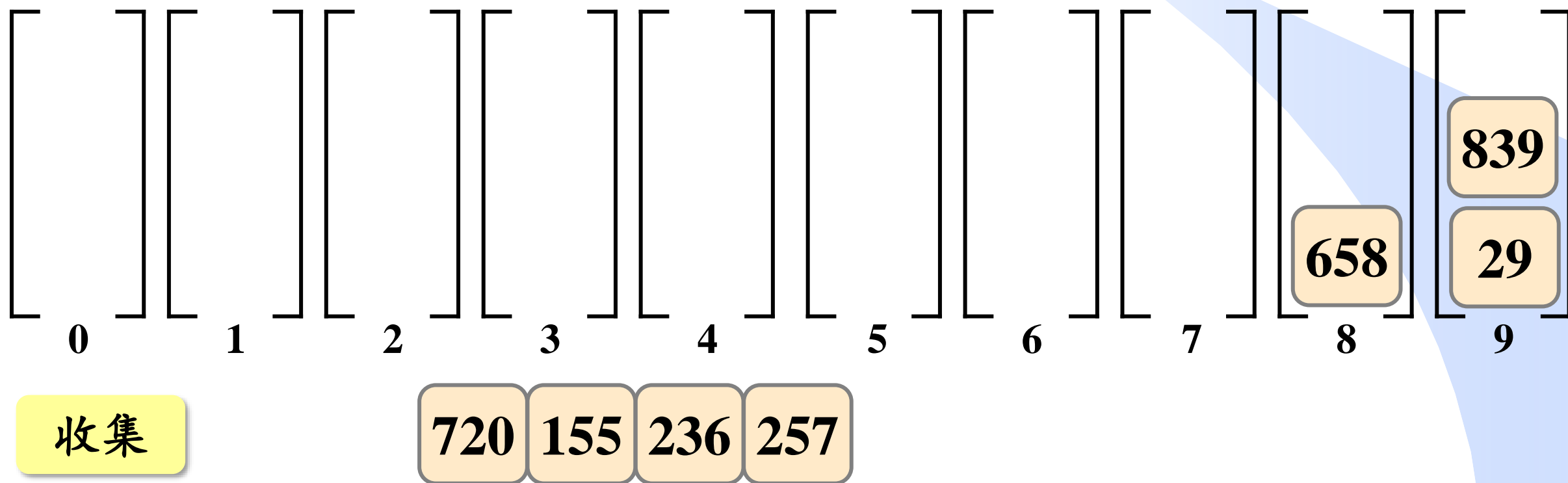
A





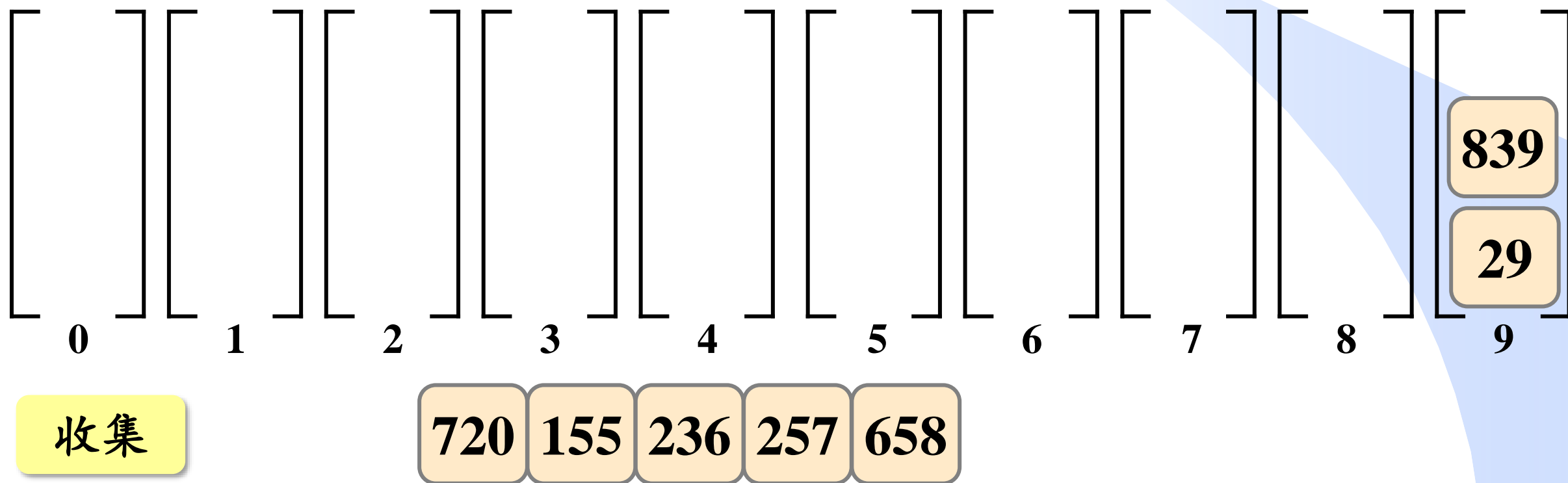
# 基于桶排序的基数排序示例：① 按个位排序

A



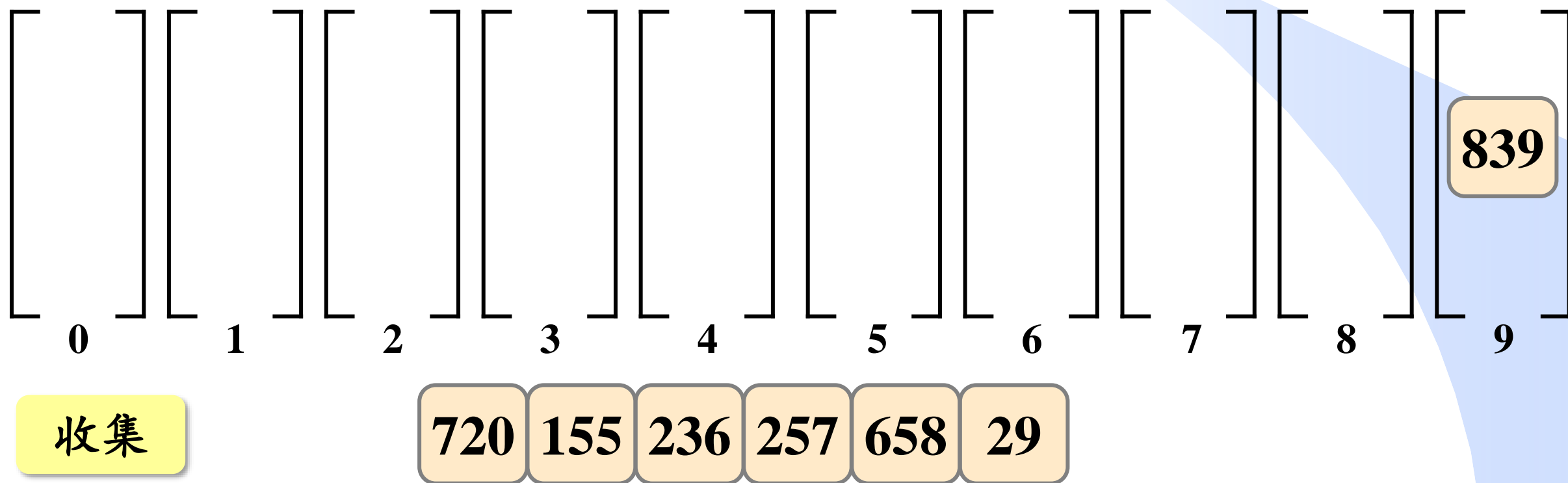
# 基于桶排序的基数排序示例：① 按个位排序

A



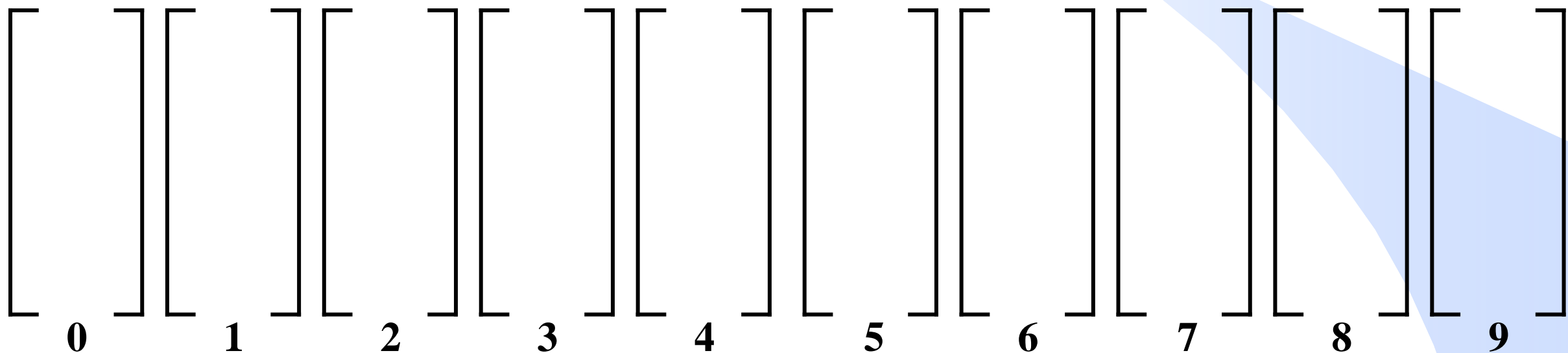
# 基于桶排序的基数排序示例：① 按个位排序

A



# 基于桶排序的基数排序示例：① 按个位排序

A



收集

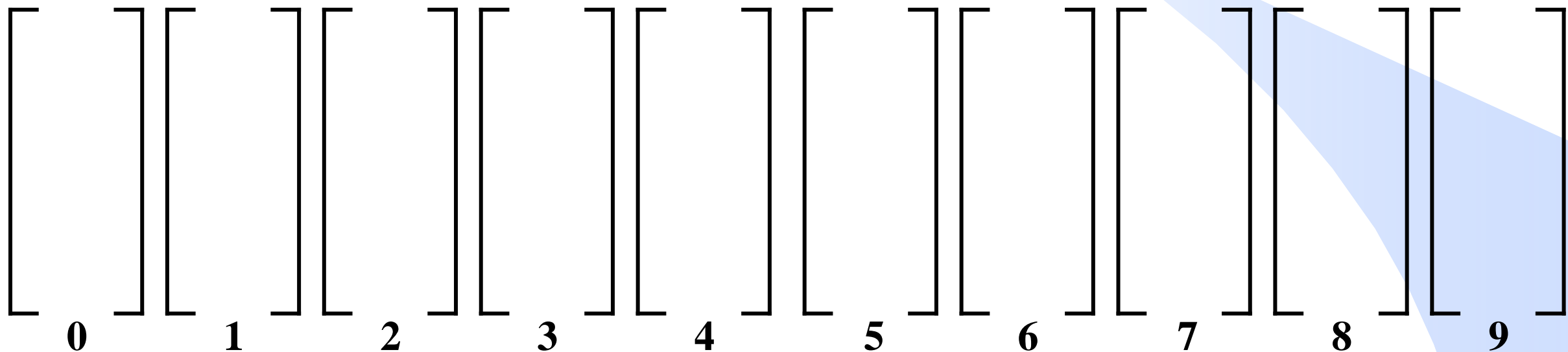
720 155 236 257 658 29 839

# 基于桶排序的基数排序示例：② 按十位排序

A

720 155 236 257 658 29 839

按十位  
分配

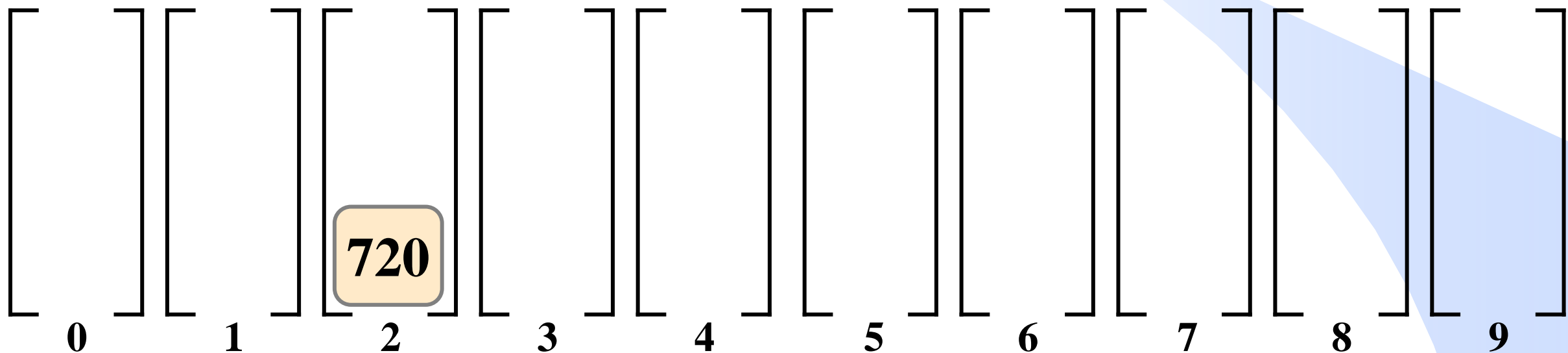


# 基于桶排序的基数排序示例：② 按十位排序

A

155 236 257 658 29 839

按十位  
分配



# 基于桶排序的基数排序示例：② 按十位排序

A

236

257

658

29

839

按十位  
分配

720

155

0

1

2

3

4

5

6

7

8

9

## 基于桶排序的基数排序示例：② 按十位排序

A

257

658

29

839

按十位  
分配

720

236

155

0

1

2

3

4

5

6

7

8

9

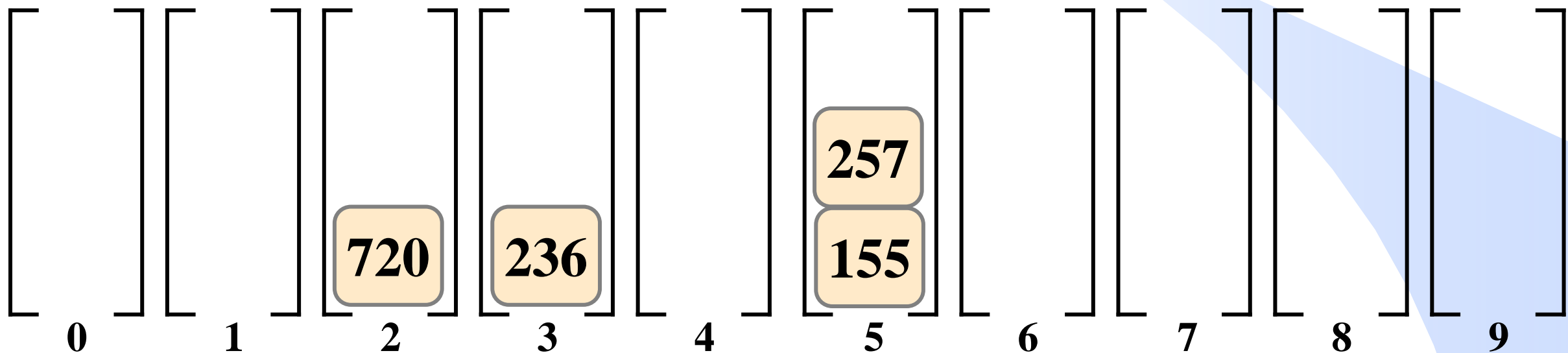


# 基于桶排序的基数排序示例：② 按十位排序

A

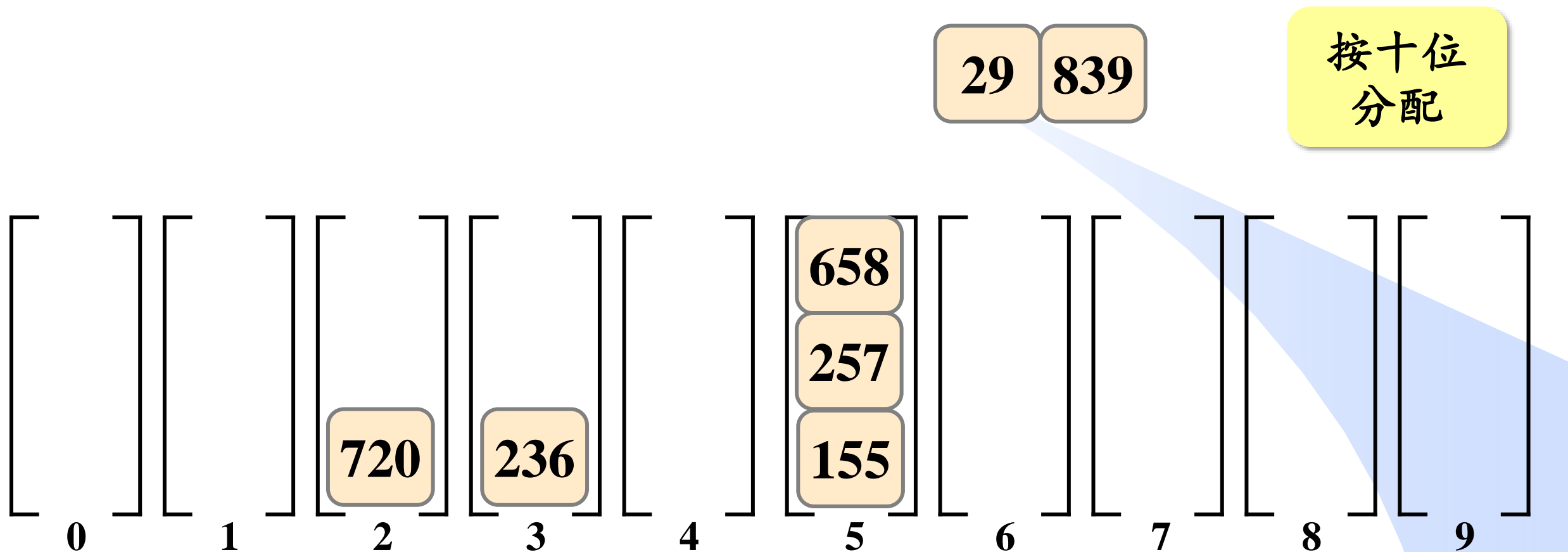
658 29 839

按十位  
分配



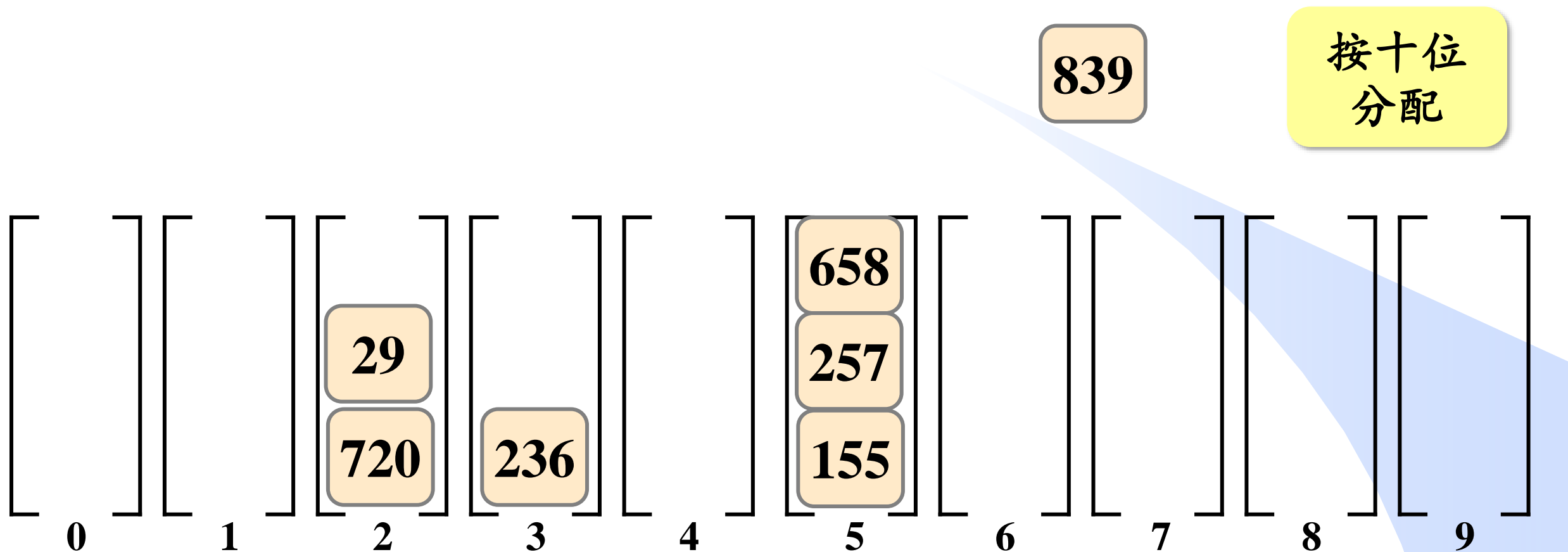
# 基于桶排序的基数排序示例：② 按十位排序

A



## 基于桶排序的基数排序示例：② 按十位排序

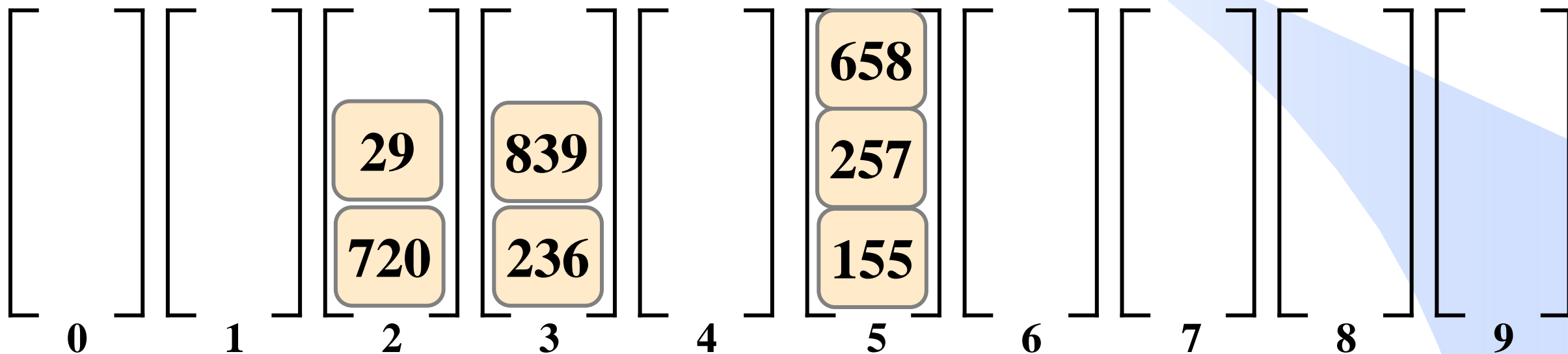
A



# 基于桶排序的基数排序示例：② 按十位排序

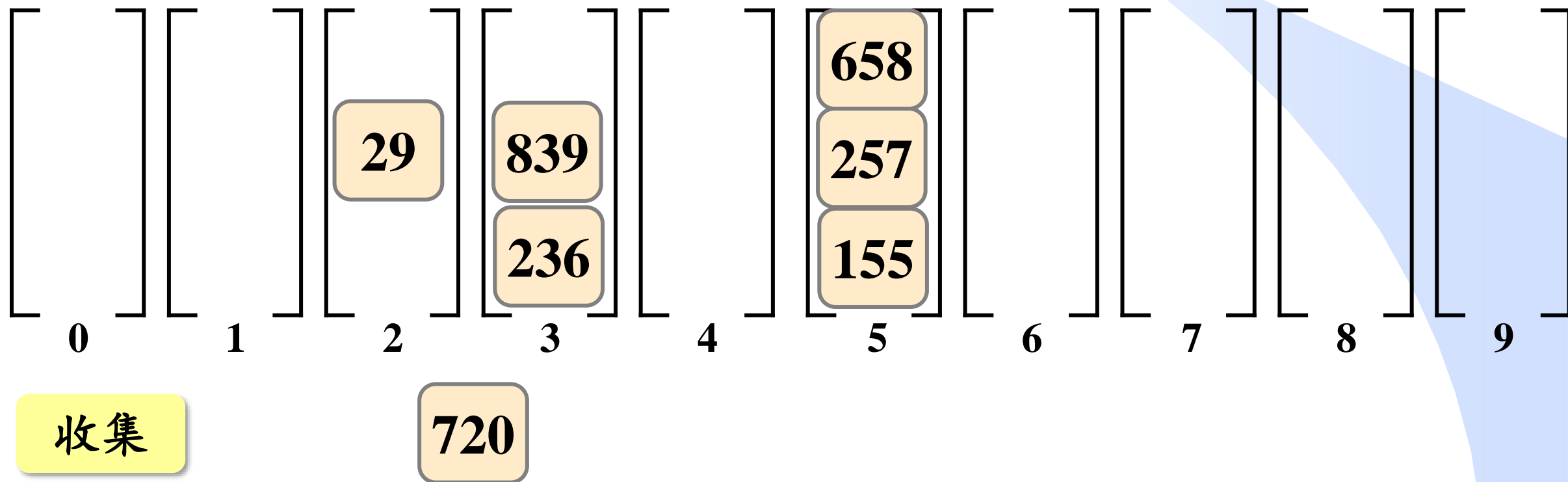
A

按十位  
分配



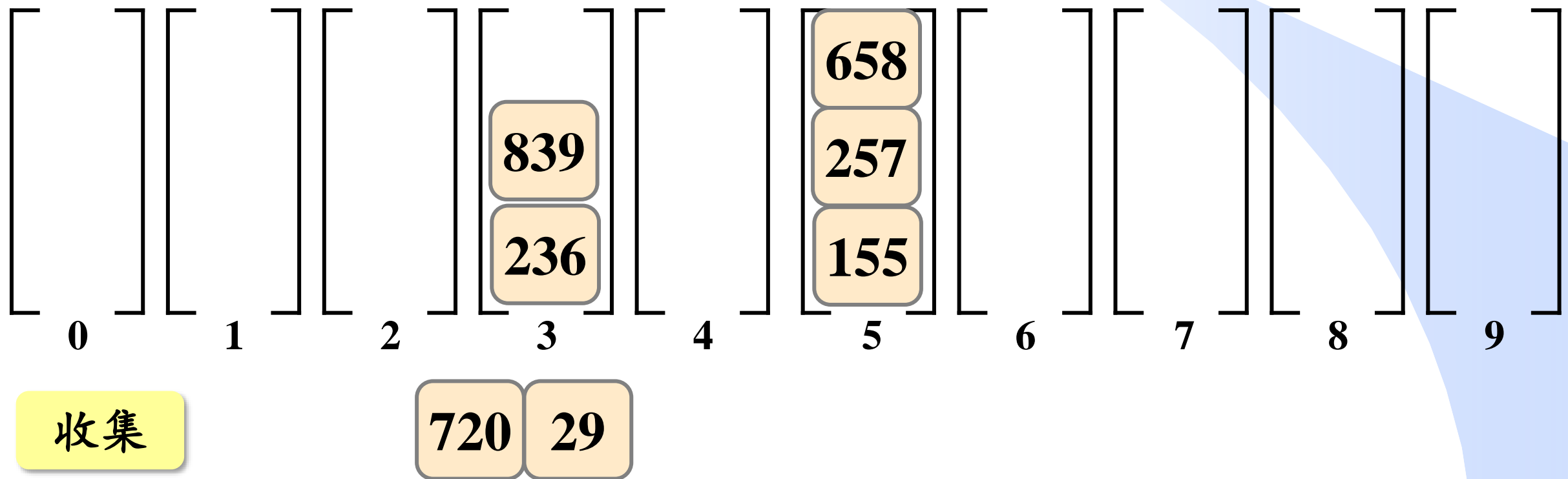
## 基于桶排序的基数排序示例：② 按十位排序

A



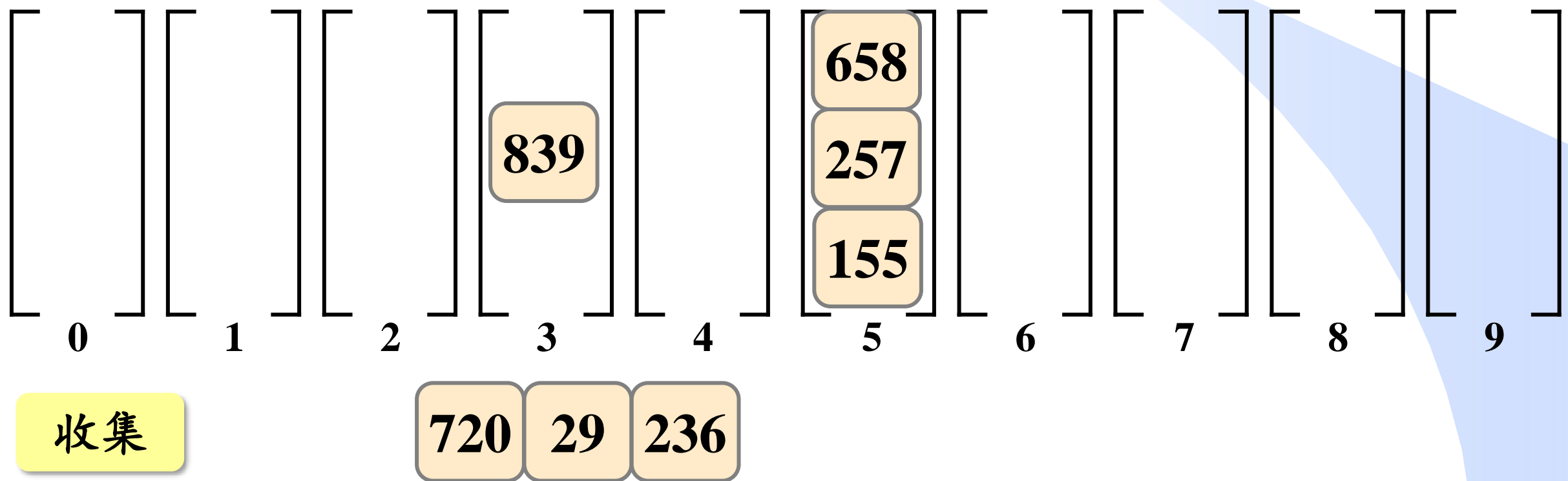
# 基于桶排序的基数排序示例：② 按十位排序

A



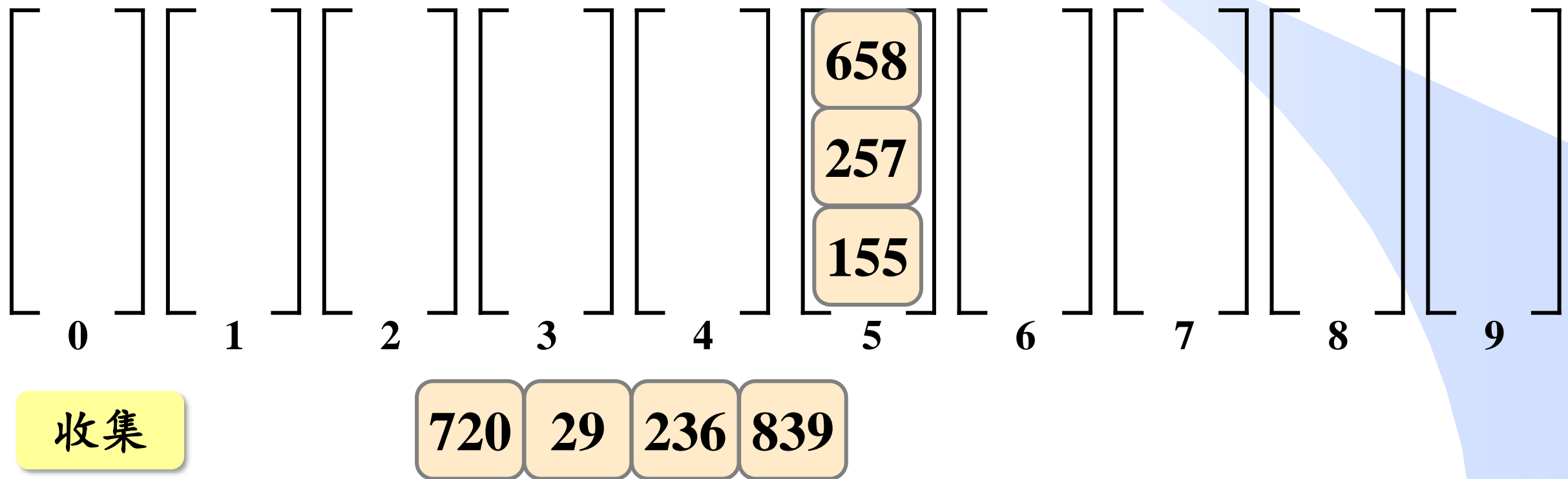
# 基于桶排序的基数排序示例：② 按十位排序

A



# 基于桶排序的基数排序示例：② 按十位排序

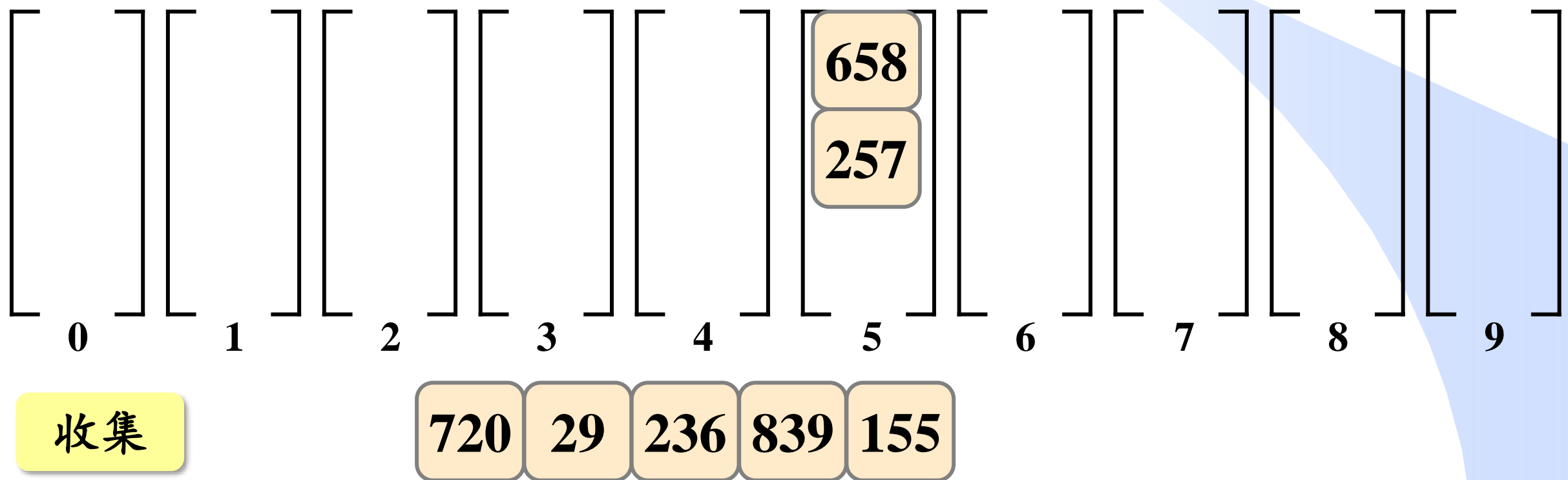
A





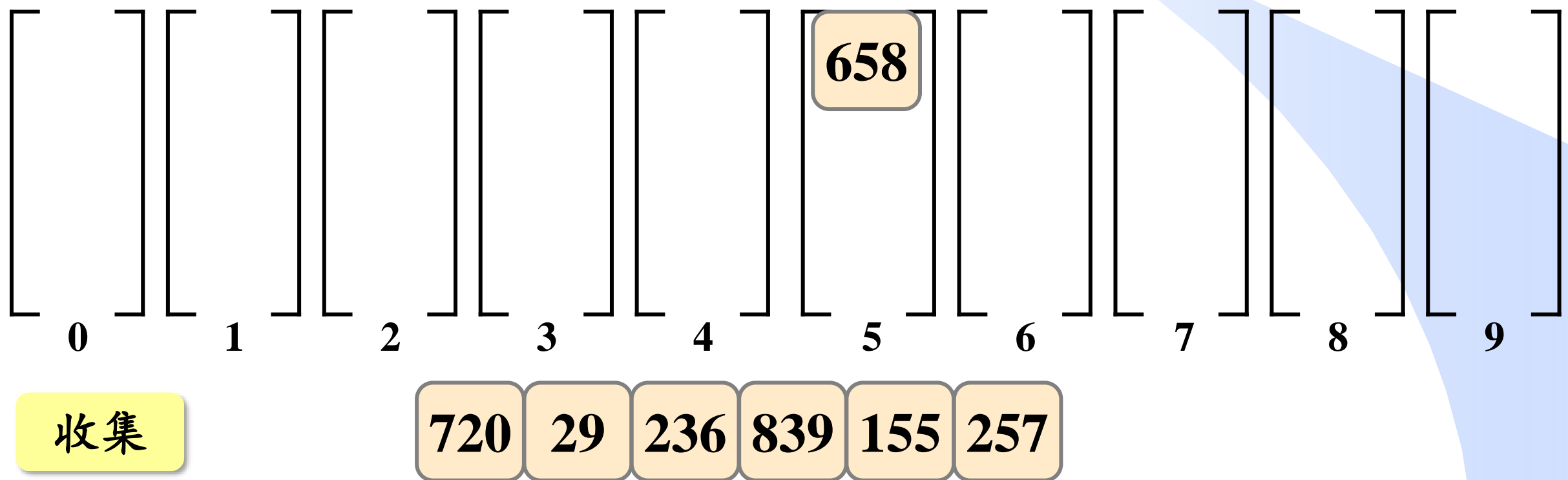
## 基于桶排序的基数排序示例：② 按十位排序

A



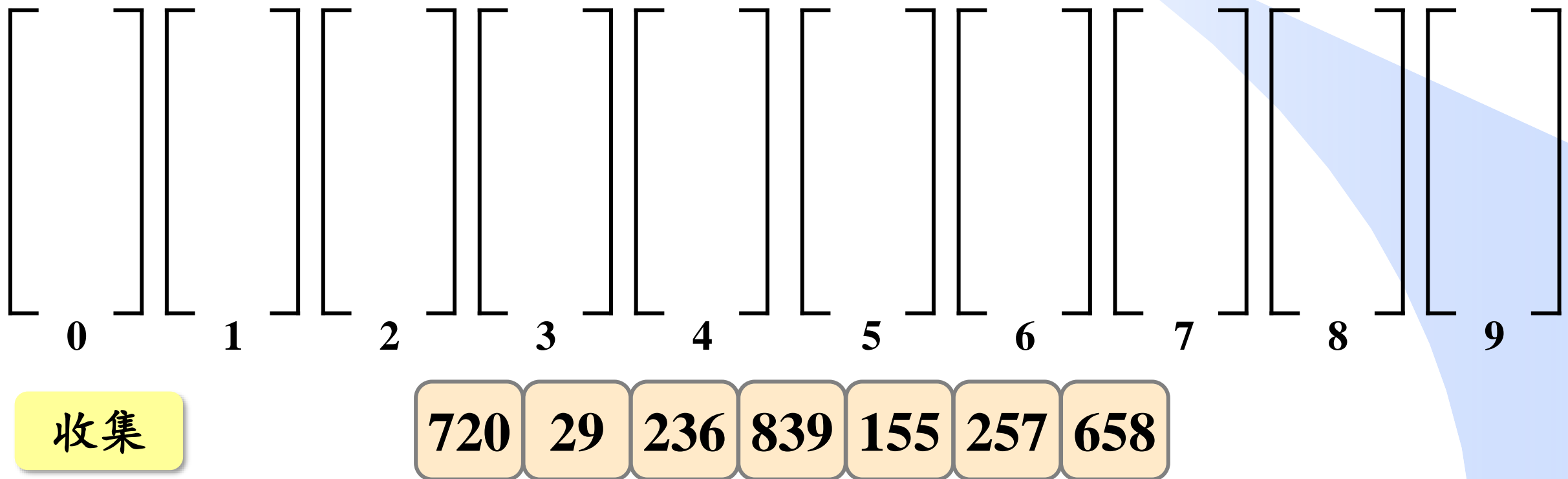
# 基于桶排序的基数排序示例：② 按十位排序

A



# 基于桶排序的基数排序示例：② 按十位排序

A

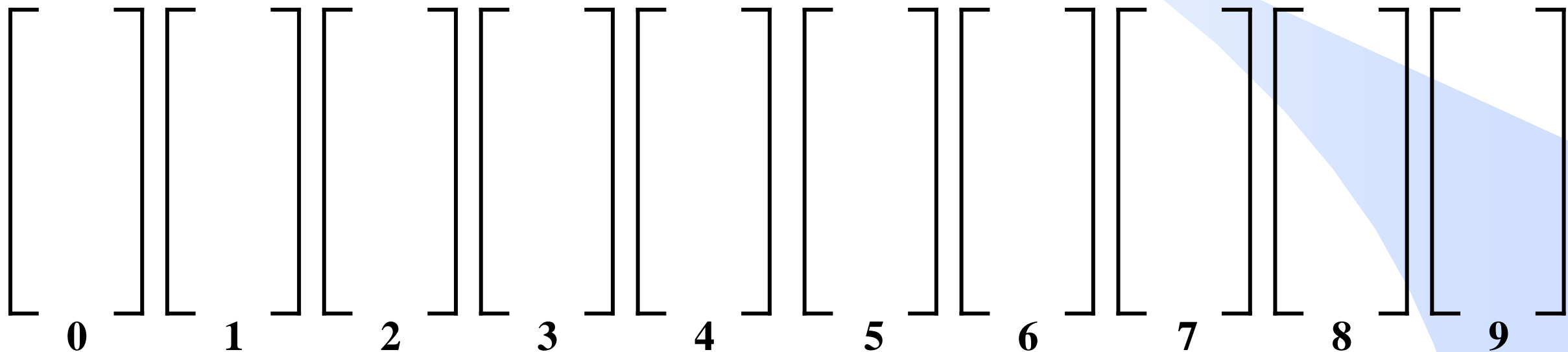


# 基于桶排序的基数排序示例：③ 按百位排序

A

720 29 236 839 155 257 658

按百位  
分配

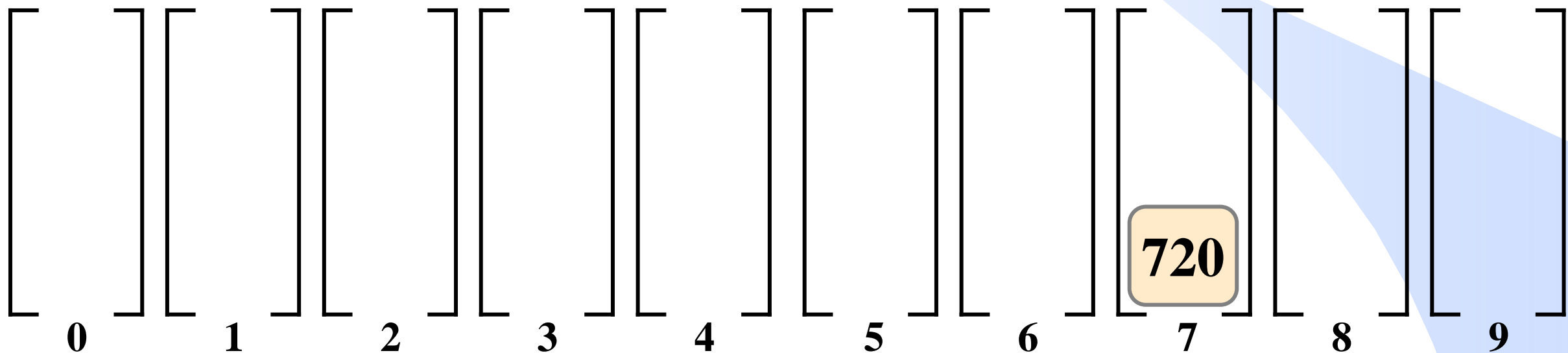


# 基于桶排序的基数排序示例：③ 按百位排序

A

29 236 839 155 257 658

按百位  
分配

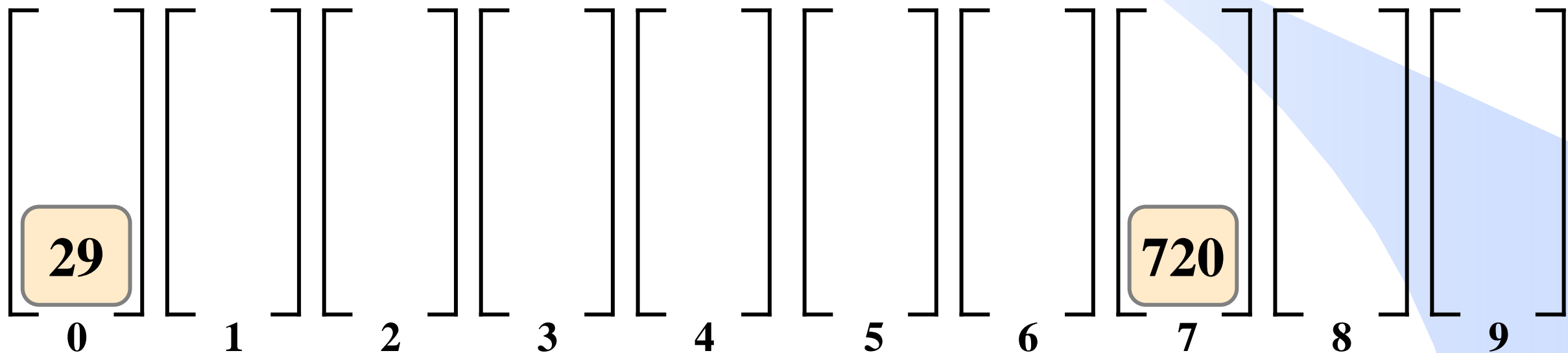


# 基于桶排序的基数排序示例：③ 按百位排序

A

236 839 155 257 658

按百位  
分配



# 基于桶排序的基数排序示例：③ 按百位排序

A

839

155

257

658

按百位  
分配

29

236

720

0

1

2

3

4

5

6

7

8

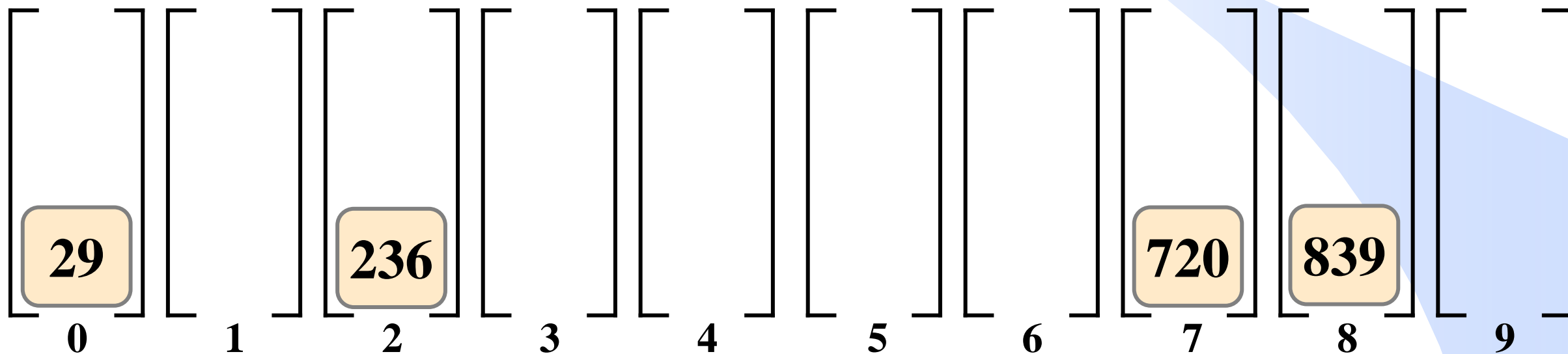
9

# 基于桶排序的基数排序示例：③ 按百位排序

A

155 257 658

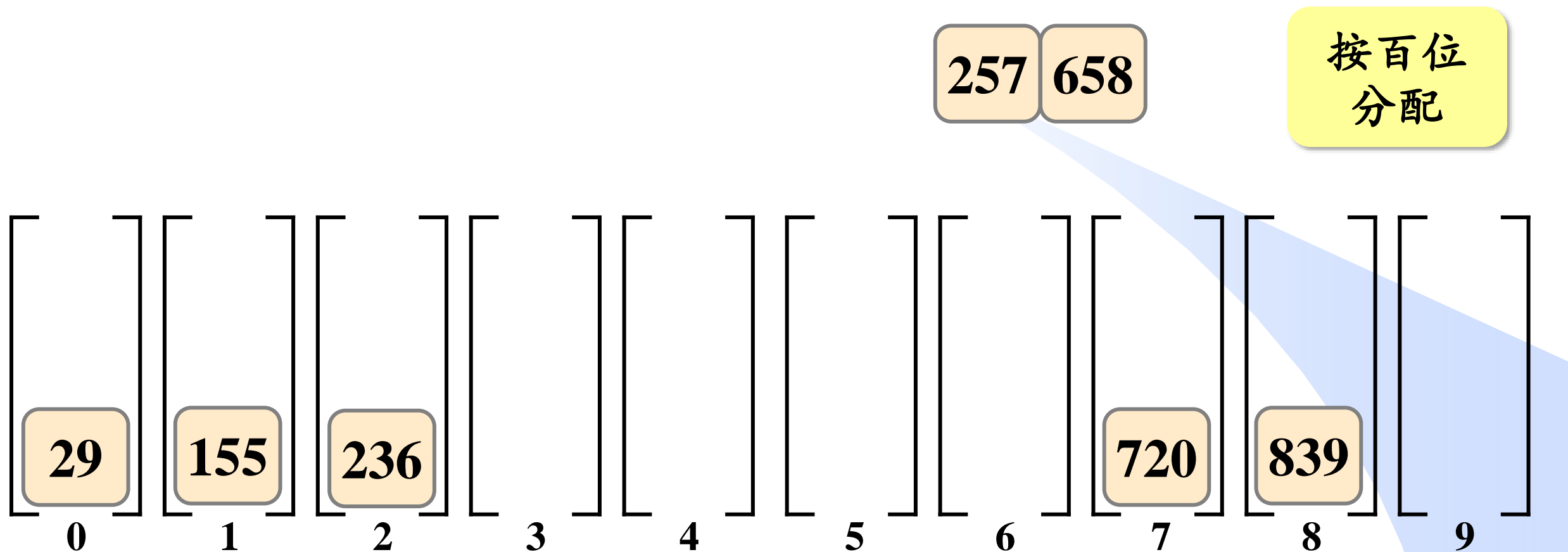
按百位  
分配





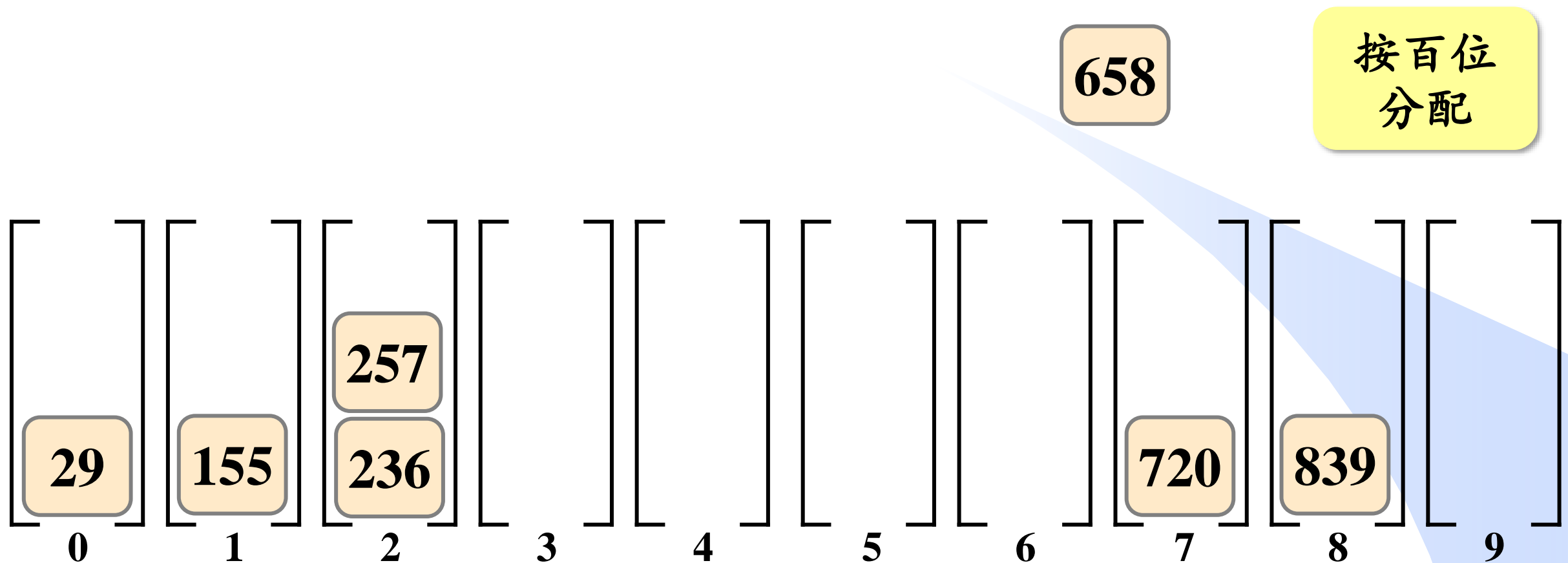
# 基于桶排序的基数排序示例：③ 按百位排序

A



# 基于桶排序的基数排序示例：③ 按百位排序

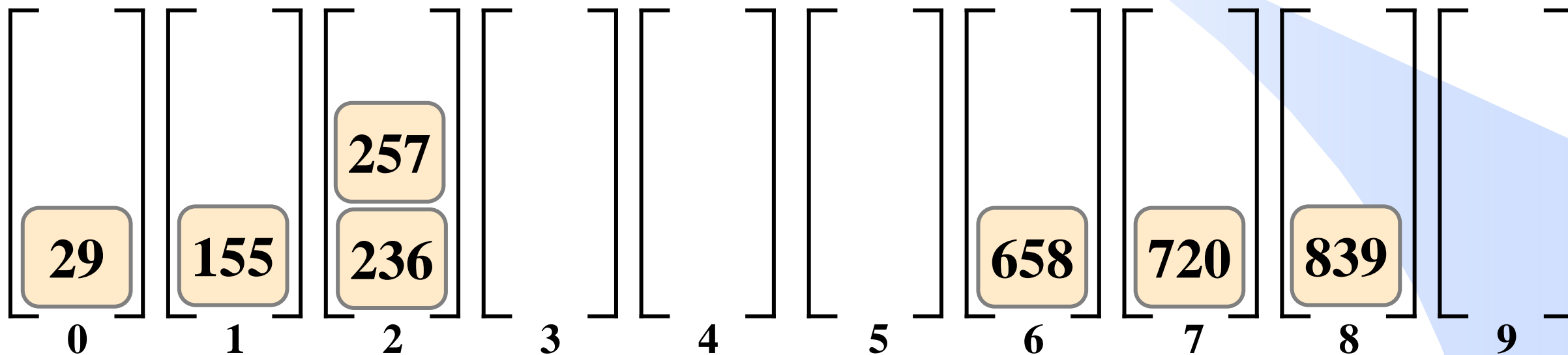
A



# 基于桶排序的基数排序示例：③ 按百位排序

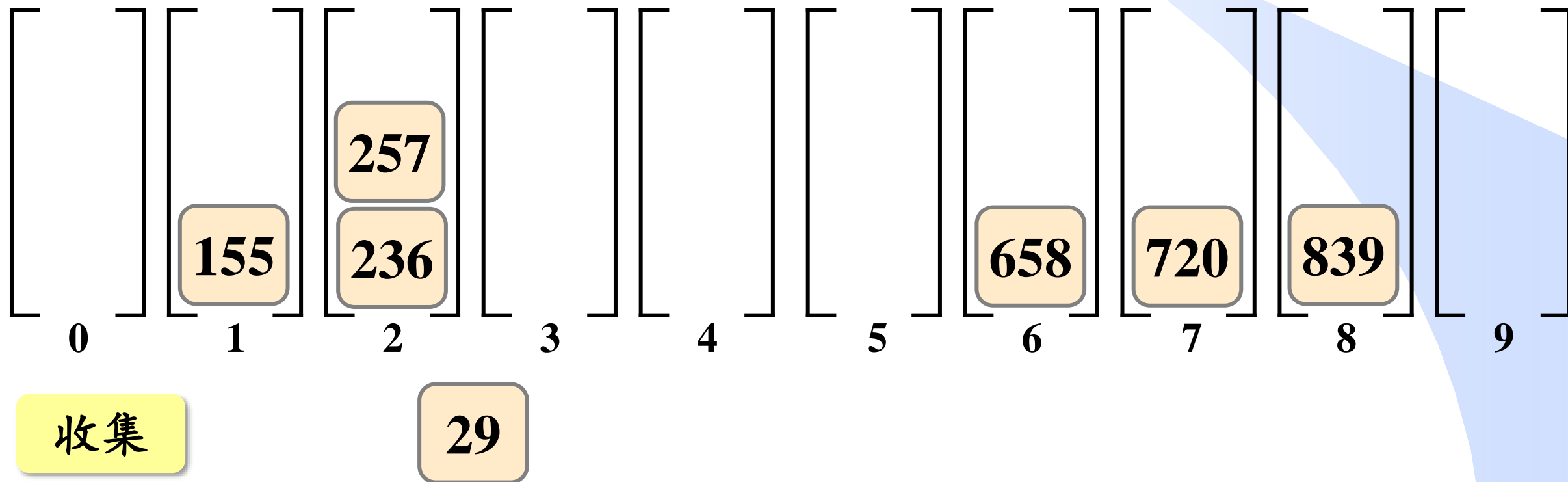
A

按百位  
分配



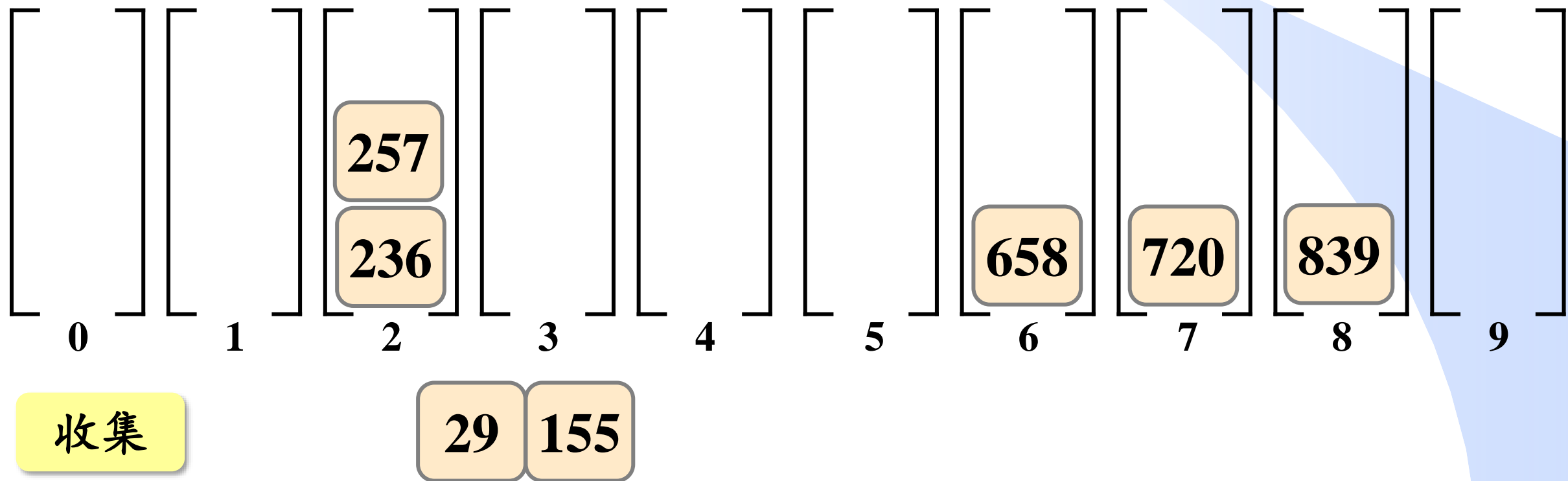
# 基于桶排序的基数排序示例：③ 按百位排序

A



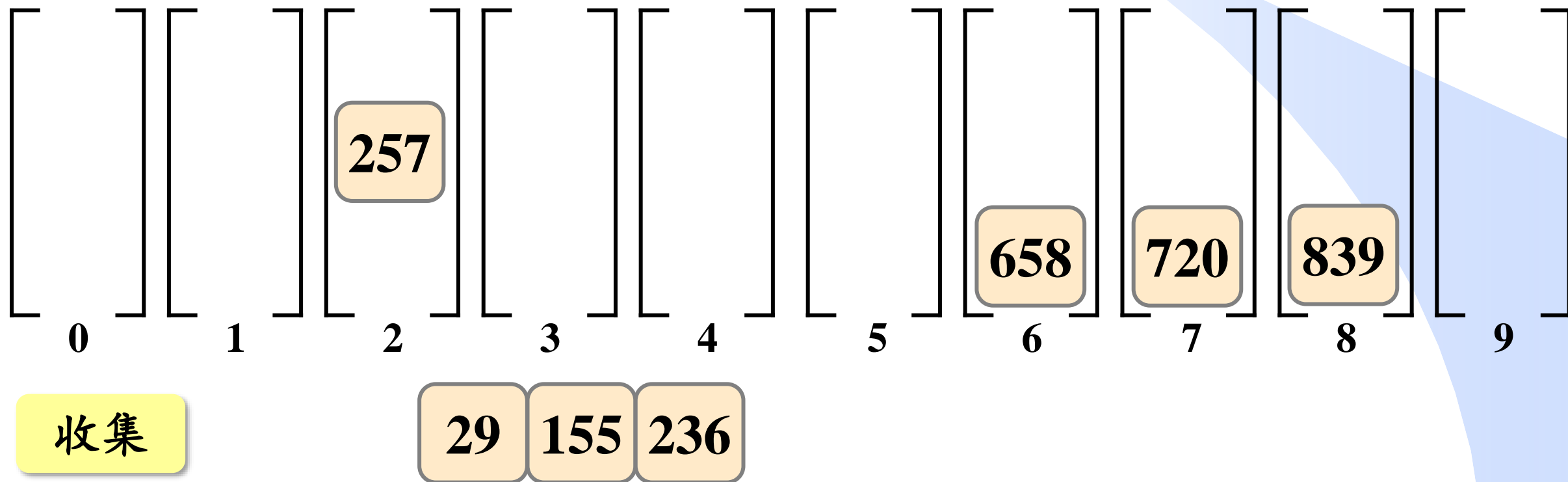
# 基于桶排序的基数排序示例：③ 按百位排序

A



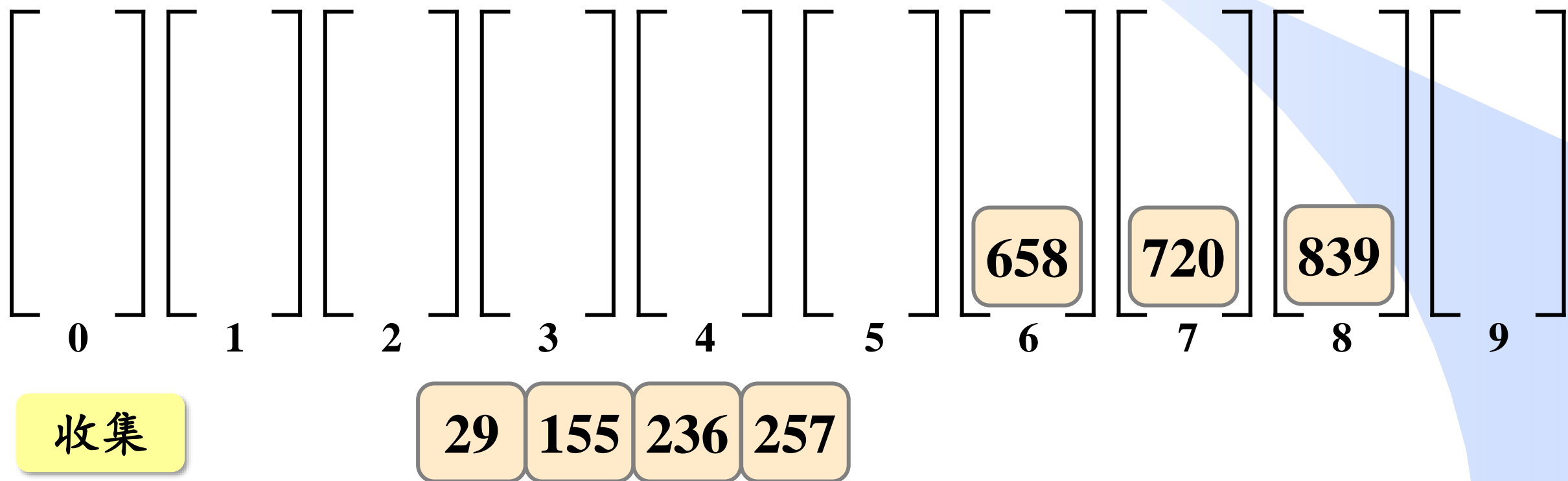
# 基于桶排序的基数排序示例：③ 按百位排序

A



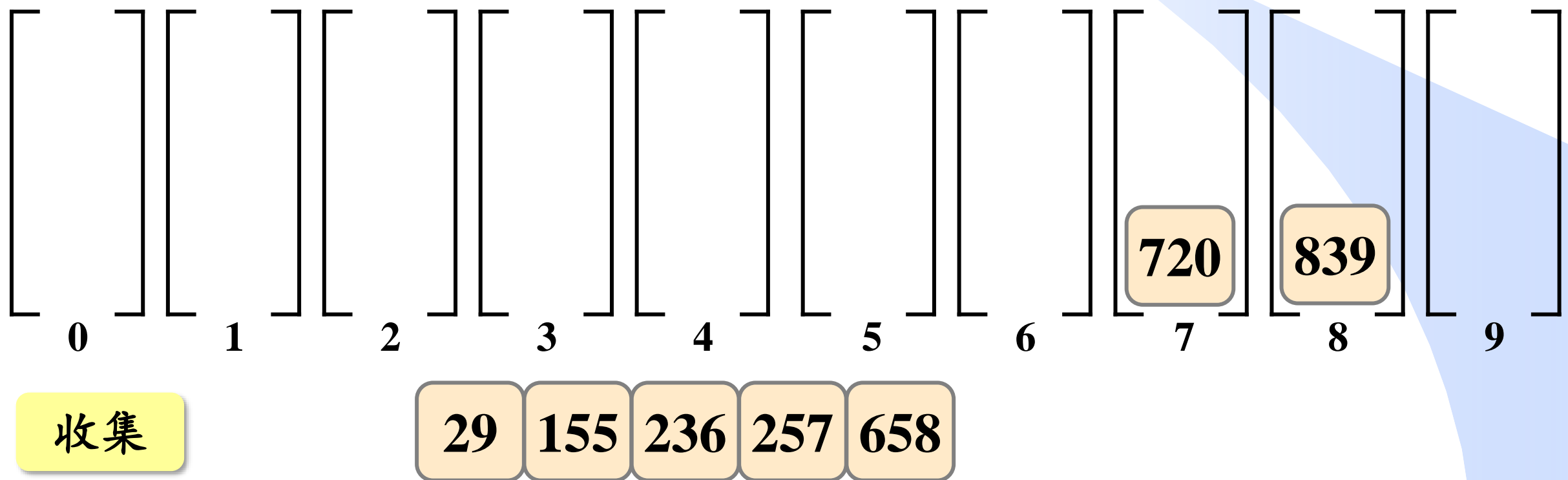
# 基于桶排序的基数排序示例：③ 按百位排序

A



# 基于桶排序的基数排序示例：③ 按百位排序

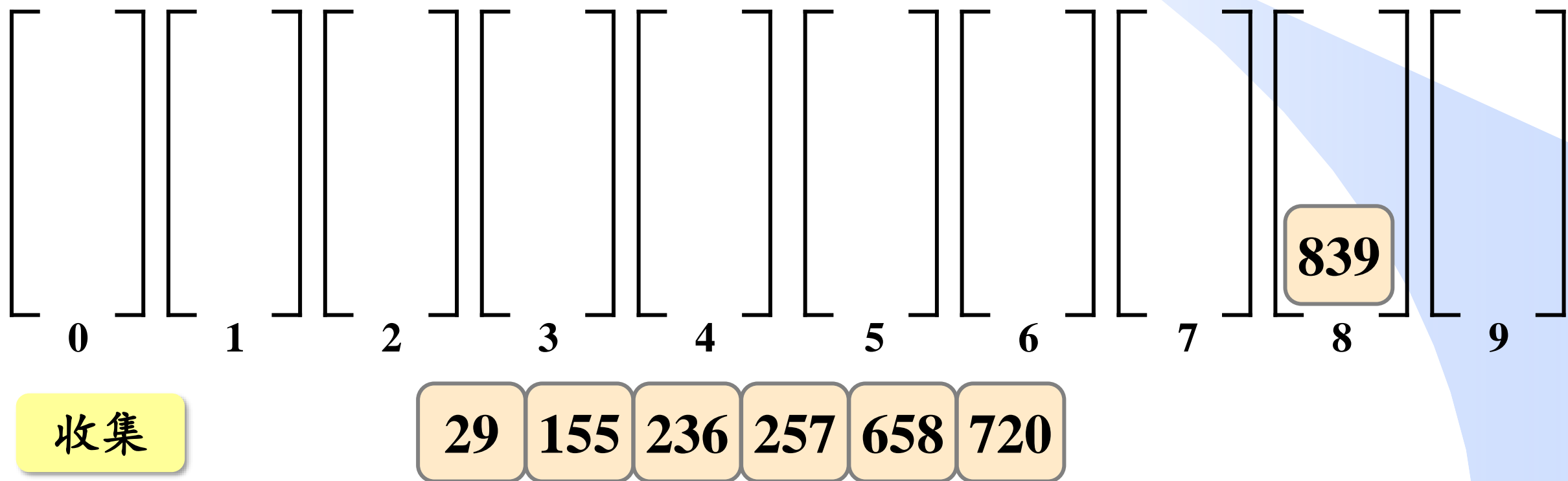
A





# 基于桶排序的基数排序示例：③ 按百位排序

A



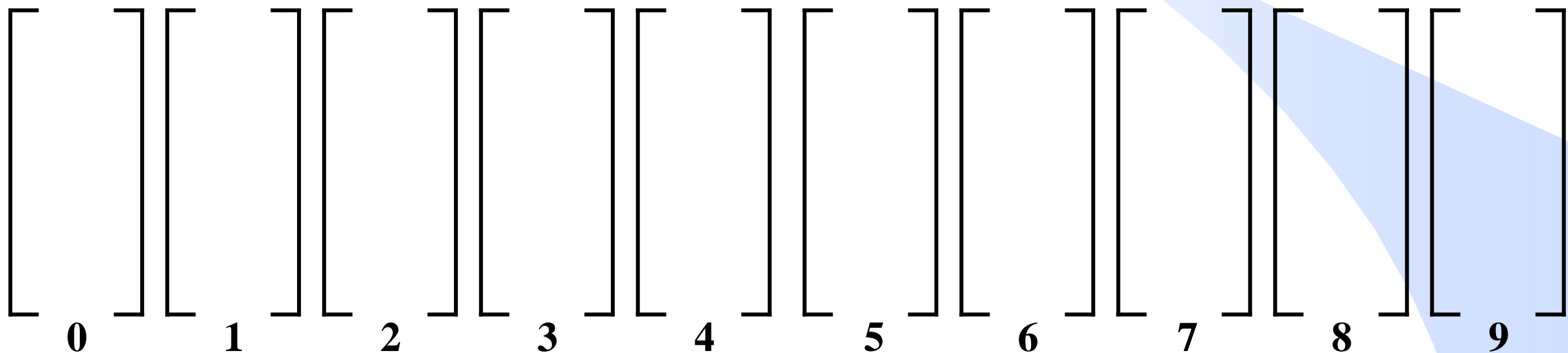
# 基于桶排序的基数排序示例：③ 按百位排序

A

稳定

时间复杂度  
 $O(d(n+r))$

空间复杂度  
 $O(n+r)$



收集

29 155 236 257 658 720 839

# 基数排序总结

排序算法	时间复杂度			空间复杂度	稳定性
	最好	平均	最坏		
基数排序	$O(d(n+r))$	$O(d(n+r))$	$O(d(n+r))$	$O(n+r)$	稳定

$d$ 为关键词的位数， $r$ 为基数  
若 $d$ 和 $r$ 为常数，则时空复杂度为线性

适合于：关键词包含固定的位数或可以拆分为多位

# 基数排序 vs 桶排序

➤ 对10个整数排序，每个整数的范围为[0, 1000)。

29 257 658 839 236 720 56 237 999 155

排序算法	参数	特点	时间
桶排序	$n=10$ $m=1000$	需1000个桶	$O(n+m)=$ $O(10+1000)$
基数排序	$n=10$ $d=3, r=10$	需10个桶，但要做3轮	$O(d(n+r))=$ $O(3(10+10))$