Sorting

Comparison-based Sorting ($\Omega(n \log n)$)

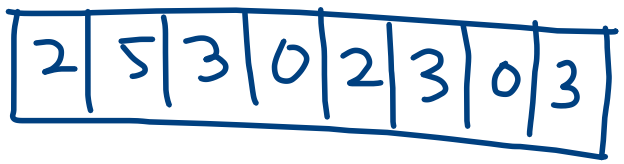
- Insertion
- Selection
- Merge
- Heap
- Quick

Sorting for special input patterns

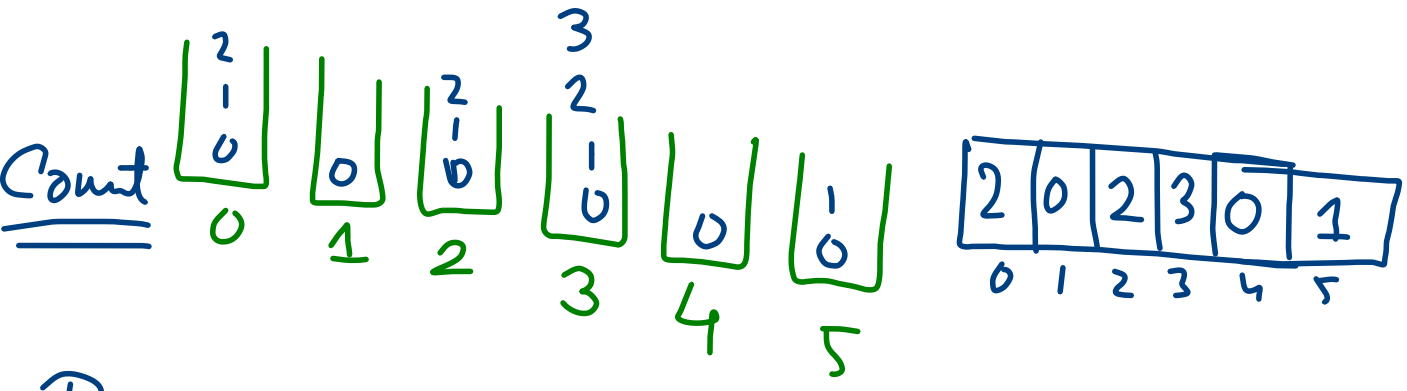
- Counting (i/p is ranged from 0 to k)
- Radix (d digit numbers with each number taking at most k-values)
- Bucket (uniformly distributed)

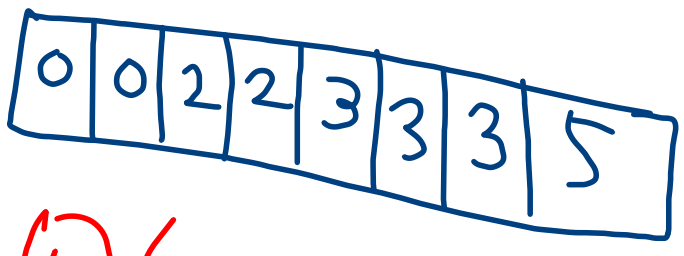
Counting Sort

Sort a sequence of integers from the range $\{0, .., k\}$

A 

$k = 5$

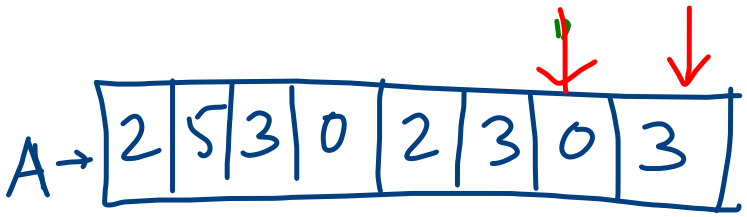
Count 

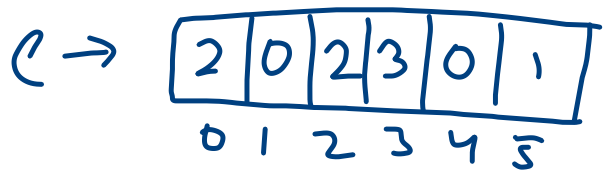
B 

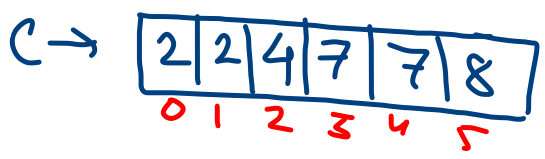
$\Theta(n+k)$

scan A

scan Count array







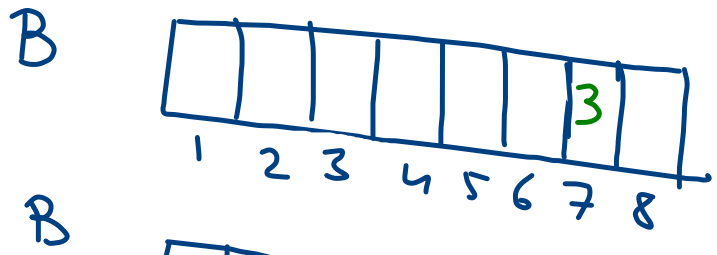
For $i = 1$ to k
$C[i] = C[i] + C[i-1]$

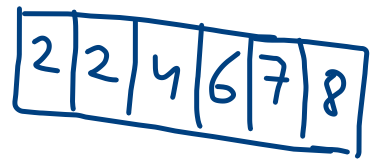
For $i = n$ to 1

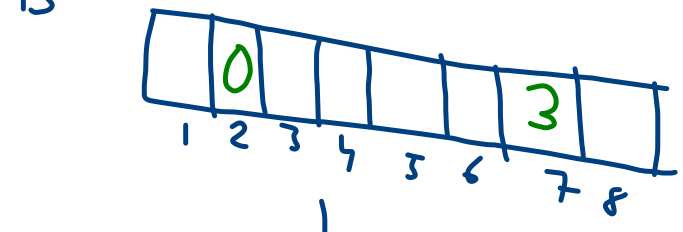
$B[UC[A[i]]] = A[i]$

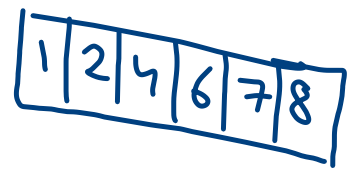
$UC[A[i]] = UC[A[i]] - 1$

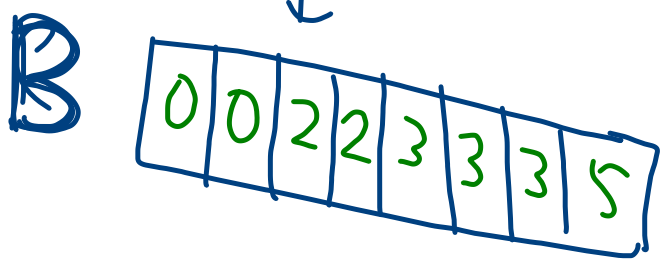
$O(n+k)$

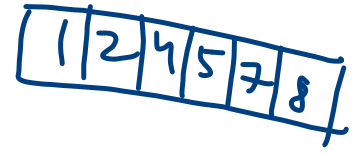










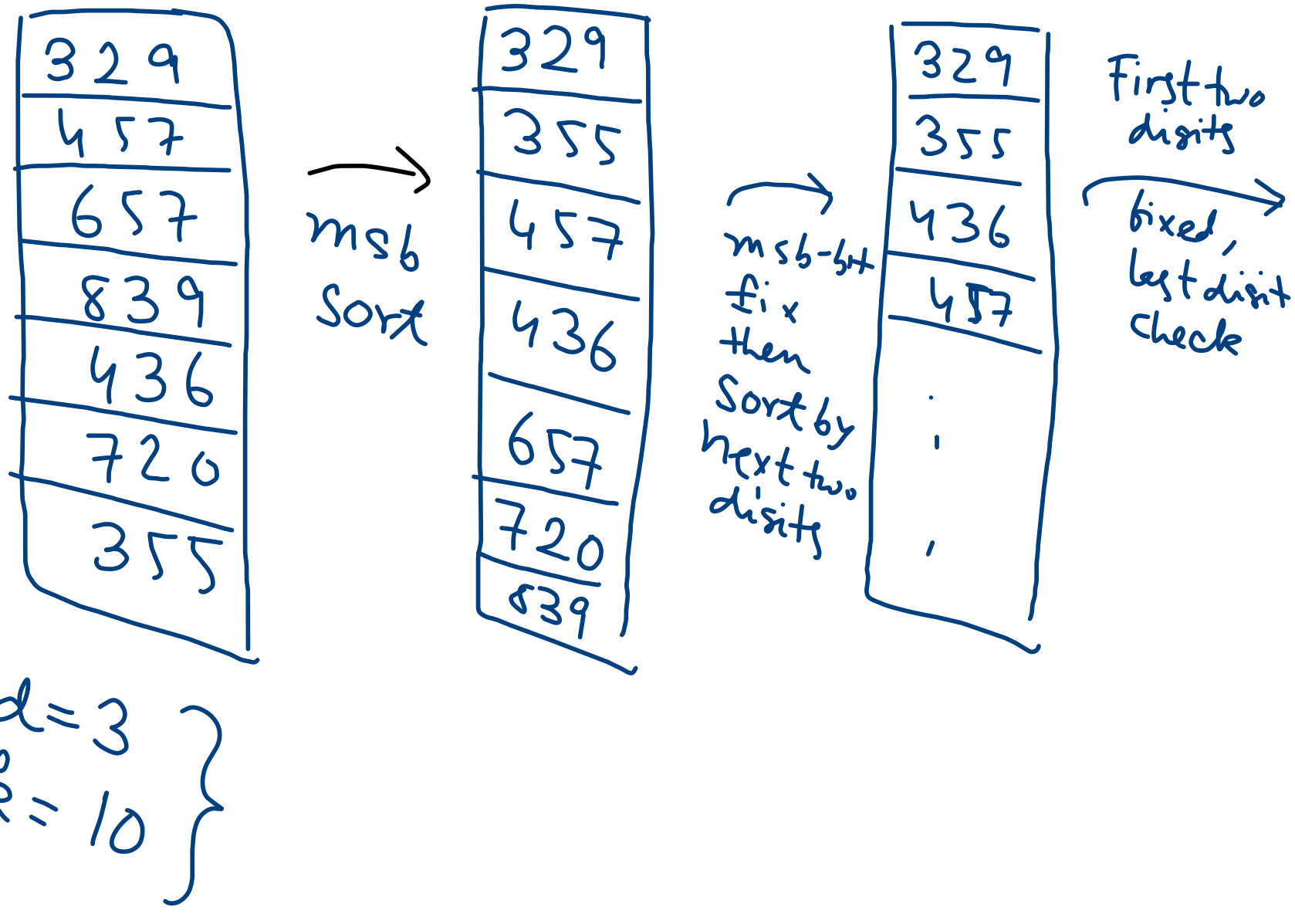


Stable

In case of same element, O/p order preserves i/p order

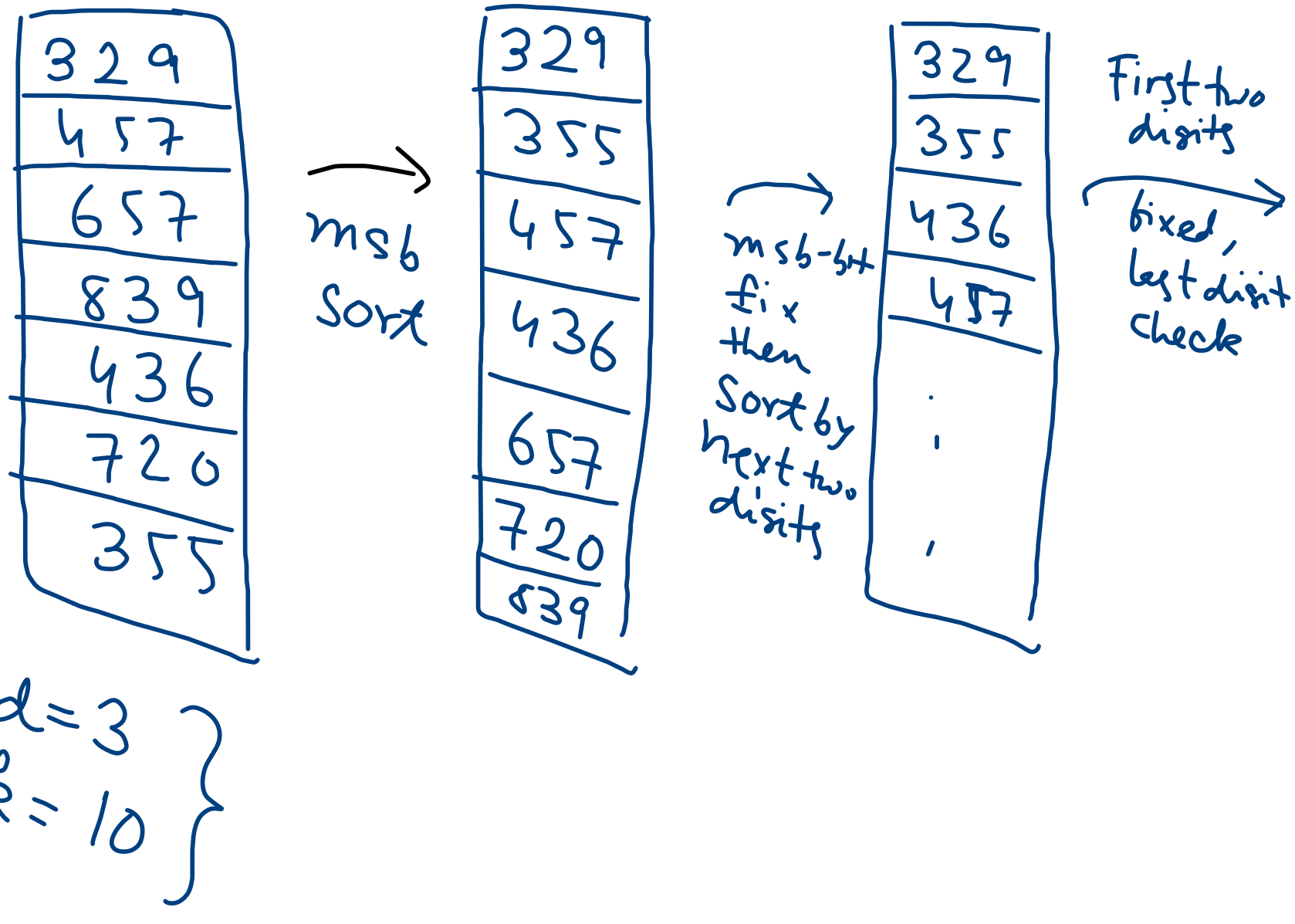
Radix Sort

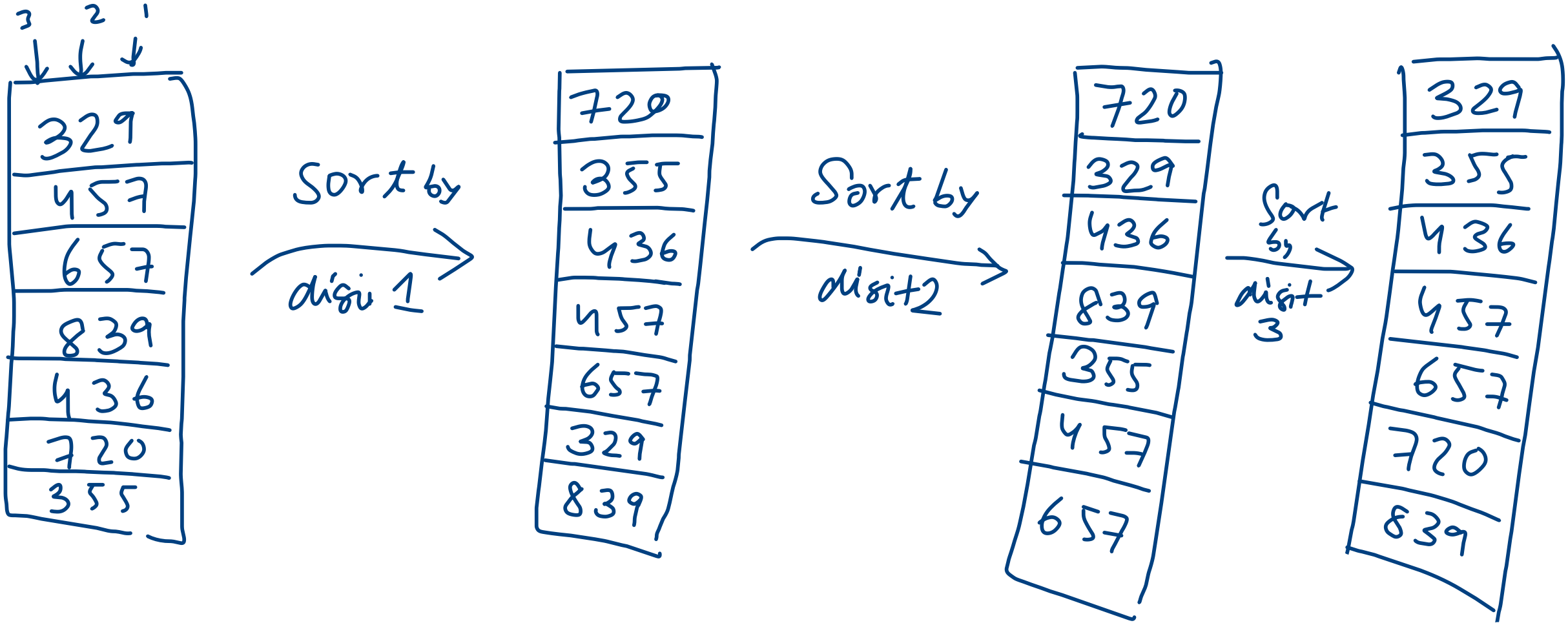
- d-digit numbers ($d \rightarrow$ constant)
- Each digit can take k-values ($k = \Theta(n)$)

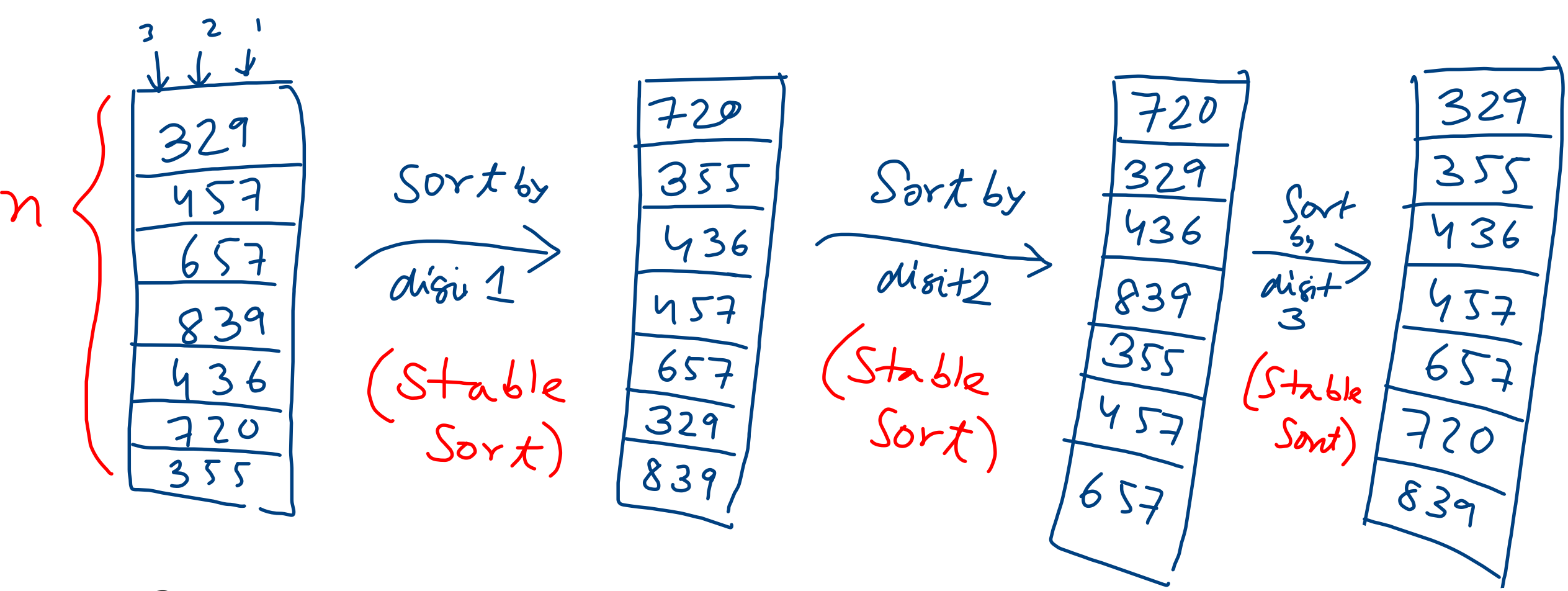


Radix Sort

- d-digit numbers ($d \rightarrow$ constant)
- Each digit can take k-values ($k = \Theta(n)$)




3 2 1
329
457
657
839
436
720
355
Sort by
digit 1
720
355
436
457
657
329
839
Sort by
digit 2
720
329
436
839
355
457
657
Sort by
digit 3
329
355
436
457
657
720
839



Radix Sort (A, d)

For $i = 1(1)\ d$

Use Counting Sort on digit i.
(Stable Sort)

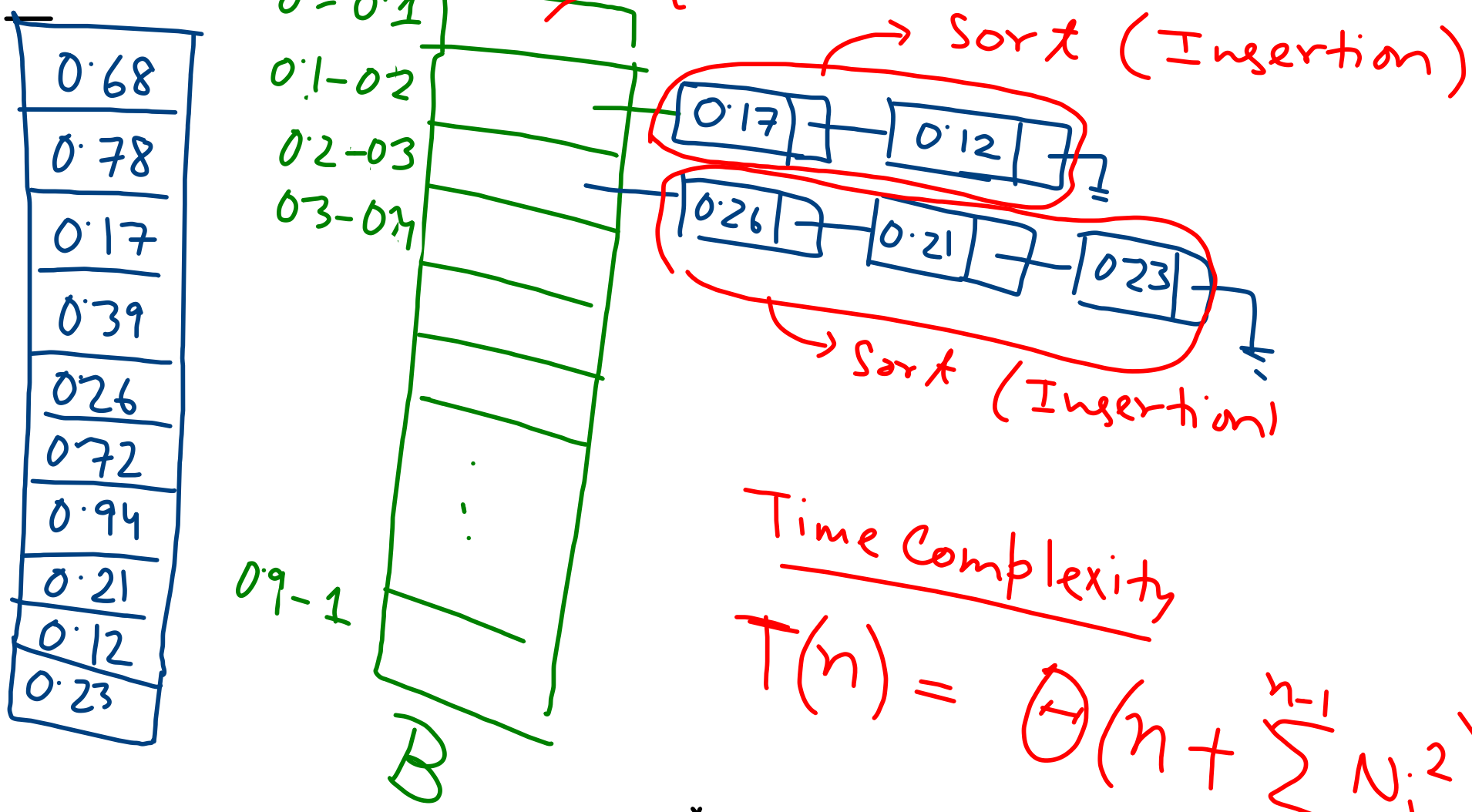
Time Complexity

$\Theta(d(n+k))$

$= \Theta(n)$ [$d \to$ constant, $k = \Theta(n)$]

Bucket Sort

- Uniformly $[0,1]$



$\Theta(n)$

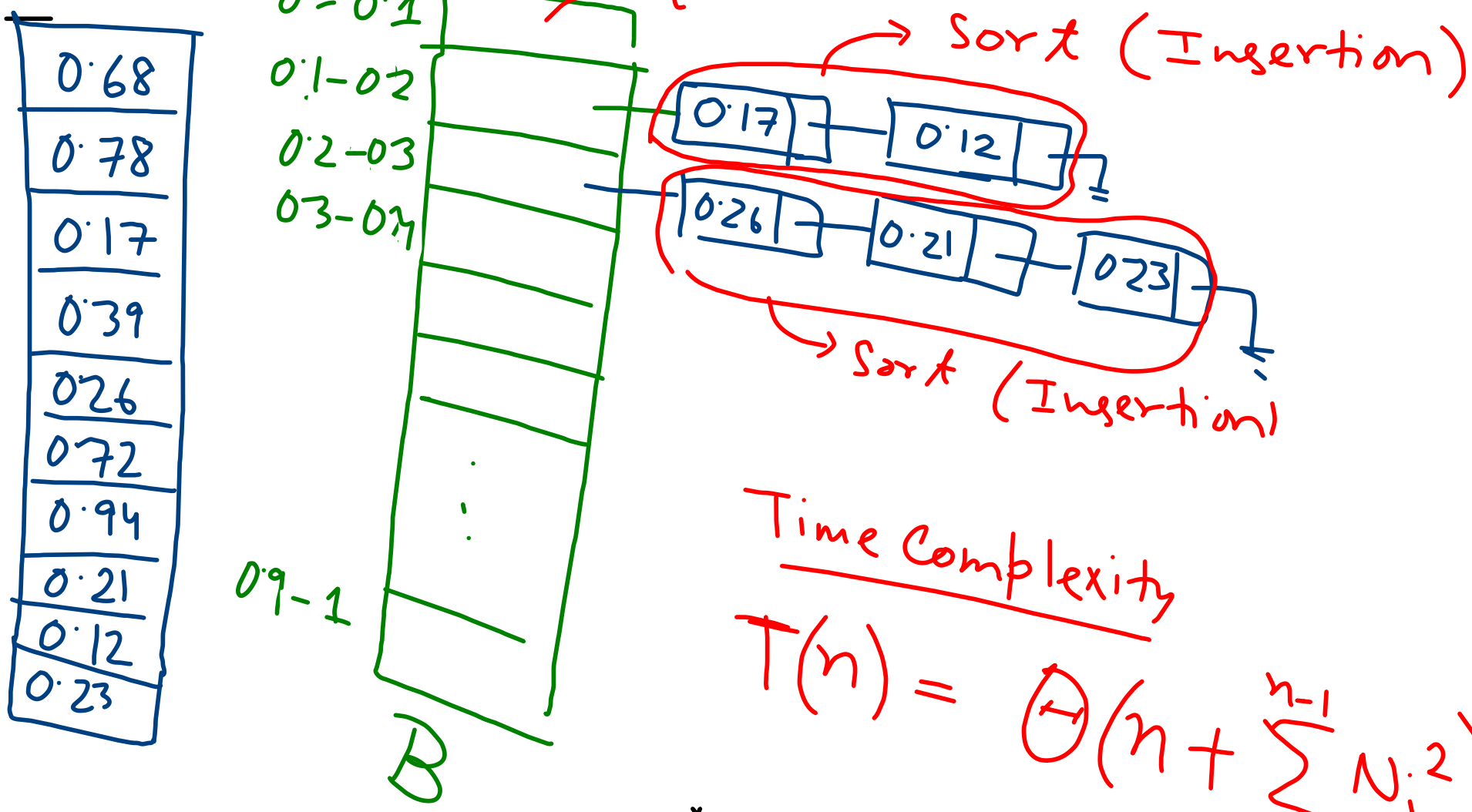
Time Complexity

$$T(n) = \Theta\left(n + \sum_{i=0}^{n-1} N_i^2\right)$$

$N_i \rightarrow$ R.V. denotes # elements in $B[i]$

Bucket Sort

- Uniformly $[0,1]$



$\Theta(n)$

Time Complexity

$$T(n) = \Theta\left(n + \sum_{i=0}^{n-1} N_i^2\right)$$

$N_i \rightarrow$ R.V. denotes # elements in $B[i]$

$$E[T(n)] = \Theta(n) + \Theta\left(\sum_{i=0}^{n-1} E[N_i^2]\right) = \Theta(n) + \Theta(n(2-\tfrac{1}{n})) = \Theta(n)$$

$$E[N_i^2]$$

$$= E\left[\left(\sum_{j=0}^{n-1} X_{ij}\right)^2\right]$$

$$= E\left[\sum_{j=0}^{n-1} X_{ij} \cdot \sum_{k=0}^{n-1} X_{ik}\right]$$

$$= \sum_{j=0}^{n-1} E[X_{ij}^2] + \sum_{j=0}^{n-1} \sum_{\substack{k=0 \\ k \neq j}}^{n-1} E[X_{ij}] \cdot E[X_{ik}]$$

$$= n \cdot \frac{1}{n} + n(n-1) \cdot \frac{1}{n^2}$$

$$= \left(2 - \frac{1}{n}\right)$$

$$\begin{cases} X_{ij} = I\{A[j] \text{ falls in bucket } i\} \\ N_i = \sum_{j=0}^{n-1} X_{ij} \end{cases}$$

$$E[X_{ij}^2] = 1^2 \cdot \frac{1}{n} + 0^2 \cdot \left(1 - \frac{1}{n}\right) = \frac{1}{n}$$

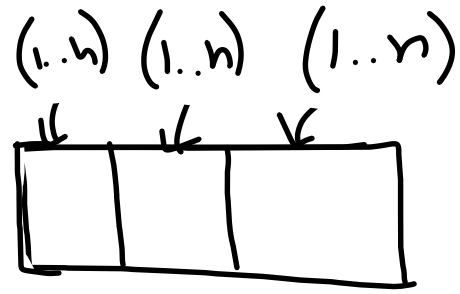
$$E[X_{ij}] \cdot E[X_{ik}] = \frac{1}{n} \cdot \frac{1}{n} = \frac{1}{n^2}$$

$j \neq k$, X_{ij} & X_{ik} are indep

Interesting Problems

Class Test (Next Tuesday)
Sorting + Order Statistics

1) Sort n integers in the range $[1, \ldots, n^3]$



Radix Sort $\rightarrow \Theta(3(n+n)) \approx \Theta(n)$

$[1, \ldots, nk] \approx \Theta(kn)$

2) In-place Algo (No extra space)

3) Algos efficient for inputs $\nearrow$ # comparisons depend on input