Linear Time Sorting

Counting Sort

↳ All the i/ps are from $\{0, \ldots, k\}$

Extra-Space is allowed.

Is this stable?

$k \rightarrow$ constant

$\Downarrow$

Time: $O(n)$, Space: $O(1)$

$k = O(n)$

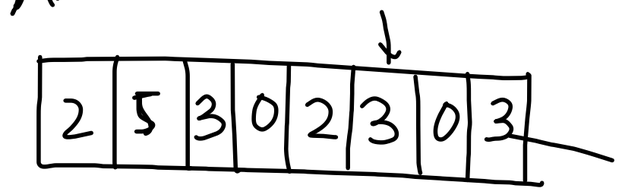
$\Downarrow$

Time: $O(n)$, Space: $O(n)$

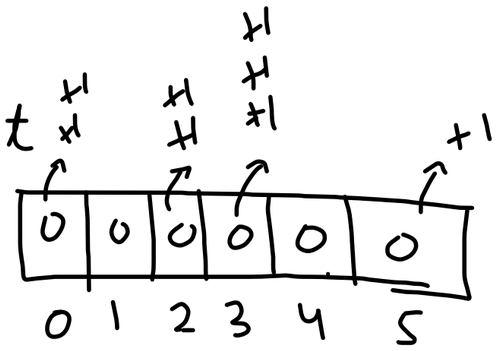
$O(n+k)$

$\{0,1,2,3,4,5\}$

A

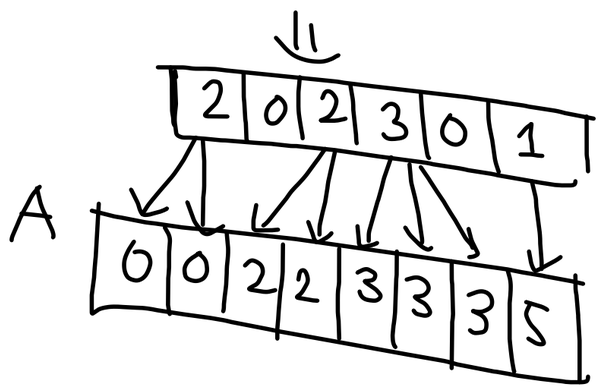


Cnt

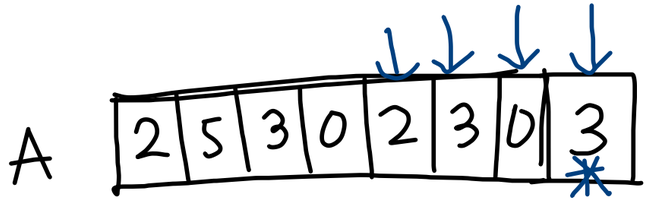


$O(k)$

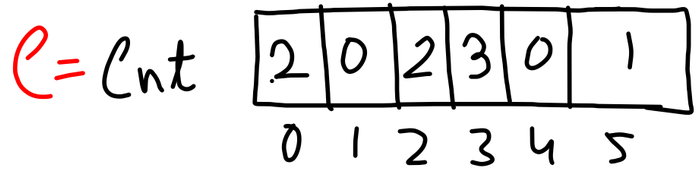
A



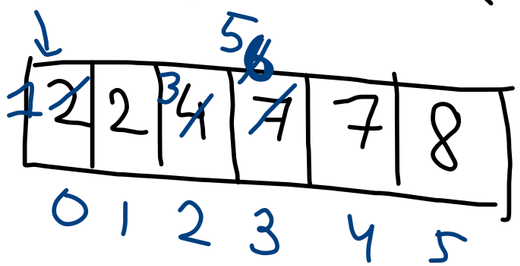
Stable Counting Sort



$k = 5$



↓ Updation (Successive Sum)



Sorted Array

$\Leftarrow O(n)$ size

Correctness $\Rightarrow$ H/W

```
For j=1(1)n
        c[A[j]] += 1

For i = 1(1) k
        c[i] = c[i] + c[i-1]

For  j = n (-1) 1
        B[c[A[j]]] = A[j];
        c[A[j]] -- = 1;
```

Radix - Sort

Suppose you have n many '3' digit numbers. How can you Sort them?

329
457
657
839
436
720
355

Counting Sort on right most digit →

720
355
436
457
657
329
839

Counting Sort on digit 2 →

720
329
436
839
355
457
657

Counting Sort on digit 3 →

329
355
436
457
657
720
839

3 2 1

Radix Sort (A, n, d)

For $i = 1(1)d$

Use Counting Sort on digit i
(Stable)

Time Complexity: $O(d(n+k))$

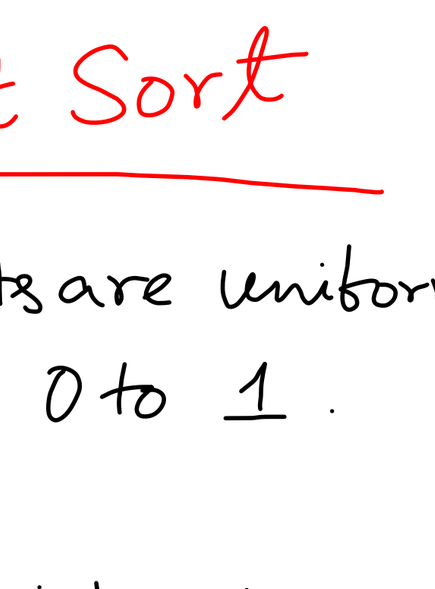
Bucket Sort

$$Exp(N_i^2) = 2 - \frac{1}{n}$$

- All the inputs are uniformly distributed between 0 to 1.

$[0,1)$ in n intervals

$n = 10$



# of element at i^{th} cell $\rightarrow N_i$

↳ Sorting $O(N_i^2)$

$$T(n) = O(n) + \sum_{i=1}^{n} O(N_i^2)$$

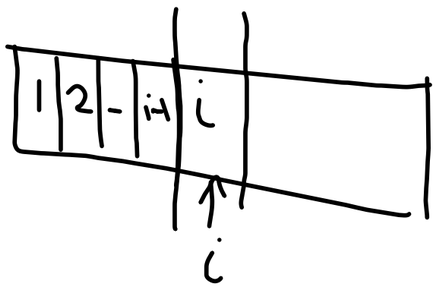
$$Exp(T(n)) = Exp\left(\sum_{i=1}^{n} O(N_i^2)\right) = O(n)$$

$A \Rightarrow \{1, 2, \ldots, n\}$

For $i = 1(1)\, n-1$

while (i != $A[i]$)

Reverse($A, i, A[i]$)



Inner Loop terminates

⇓

Proof

5 3 2 1 4

↓ Rev(A, 1, 5)

4 1 2 3 5

↓ Rev(A, 1, 4)

3 2 1 4 5

↓ Rev(A, 1, 3)

1 2 3 4 5

5 10 1 2 6 - -

6 2 1 10 5

---

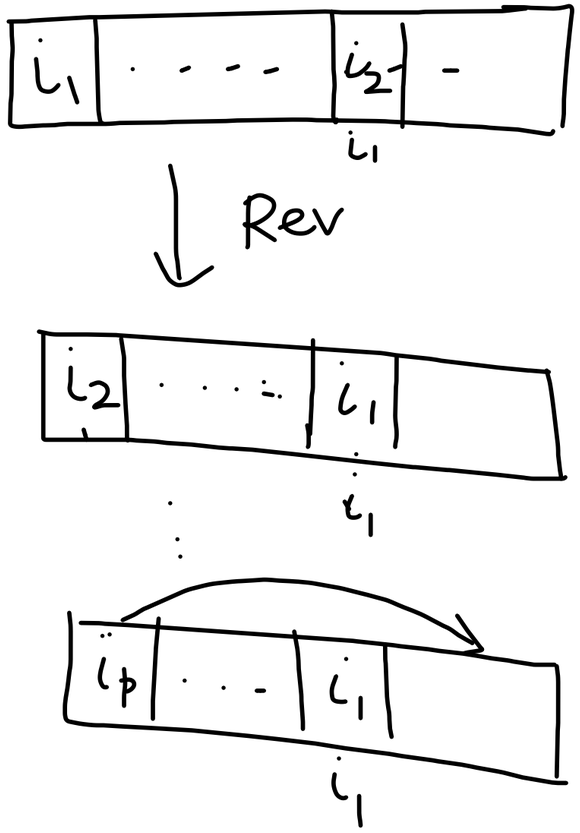
2 4 1 3

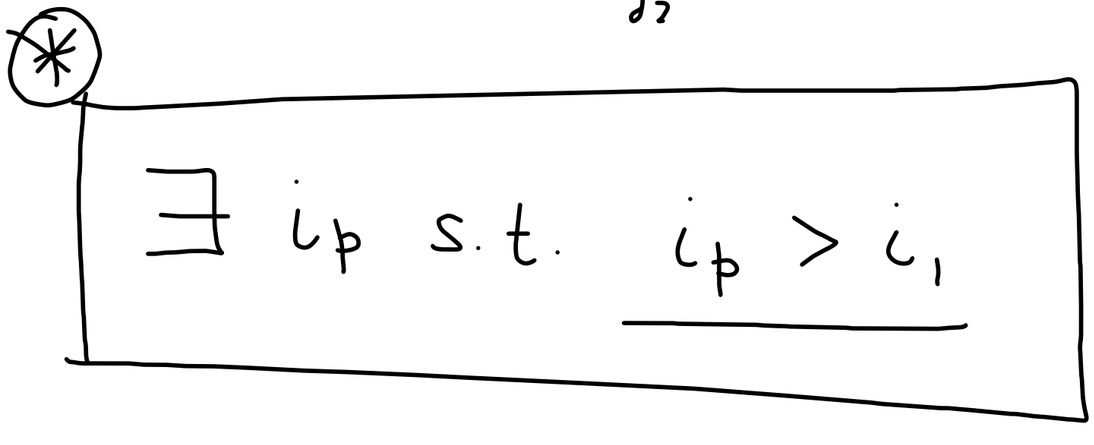
* 4 2 1 3

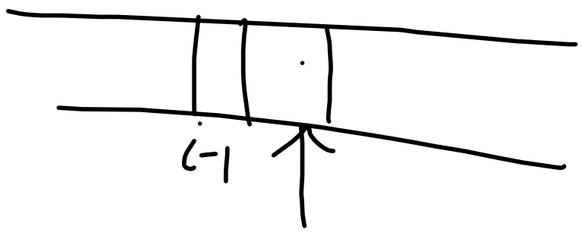
3 1 2 4

2 1 3 4

$$\begin{cases} i < j \\ \text{Rev}(i,j) \end{cases}$$







j_1
j_2
j_1
j_2

$(n-i)$

$\{ j_1, j_2, \cdots, j_t \}$

max ↓

j_q