

Insert in correct pos  
in sorted array.

Insertion Sort

```

for (j=1; j < N; j++) {
    j = i-1;
    while (j >= 0 && A[j] > A[j+1]) {
        swap(A[j], A[j+1]);
        j--;
    }
}

```

TC :  $O(N^2)$   
SC :  $O(1) \Leftarrow$

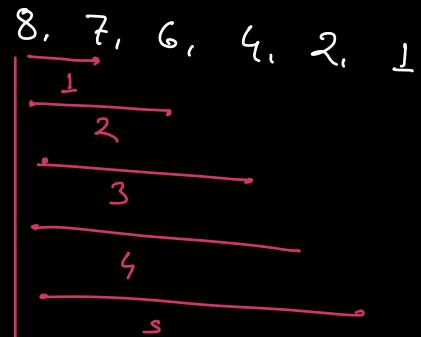
$$\begin{matrix} j & i \\ \downarrow & \downarrow \\ 0 & 1 \\ 2 & 2 \end{matrix}$$

Stable : Yes

Inplace : Yes

# swaps :  $O(N^2)$   
(WC)

1, 2, 3, 4, 5, 6, 10  
 $O(N)$



$$\begin{aligned}
& 1 + 2 + 3 + 4 + 5 + \dots + (N-1) \\
\Rightarrow & \frac{N \times (N-1)}{2}
\end{aligned}$$

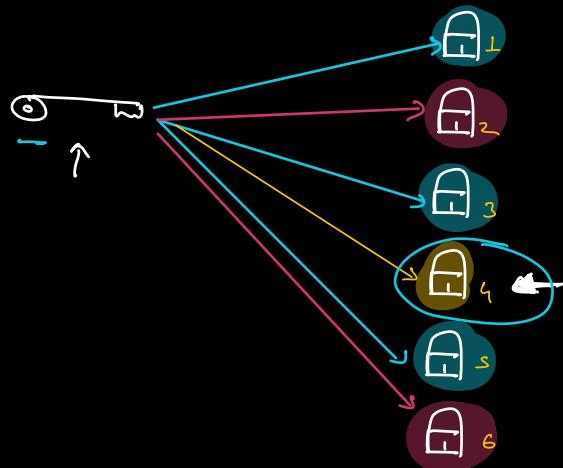
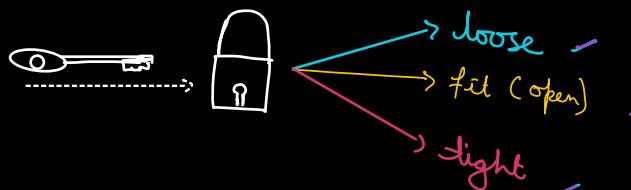
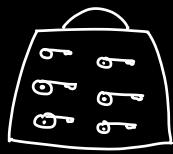
Google

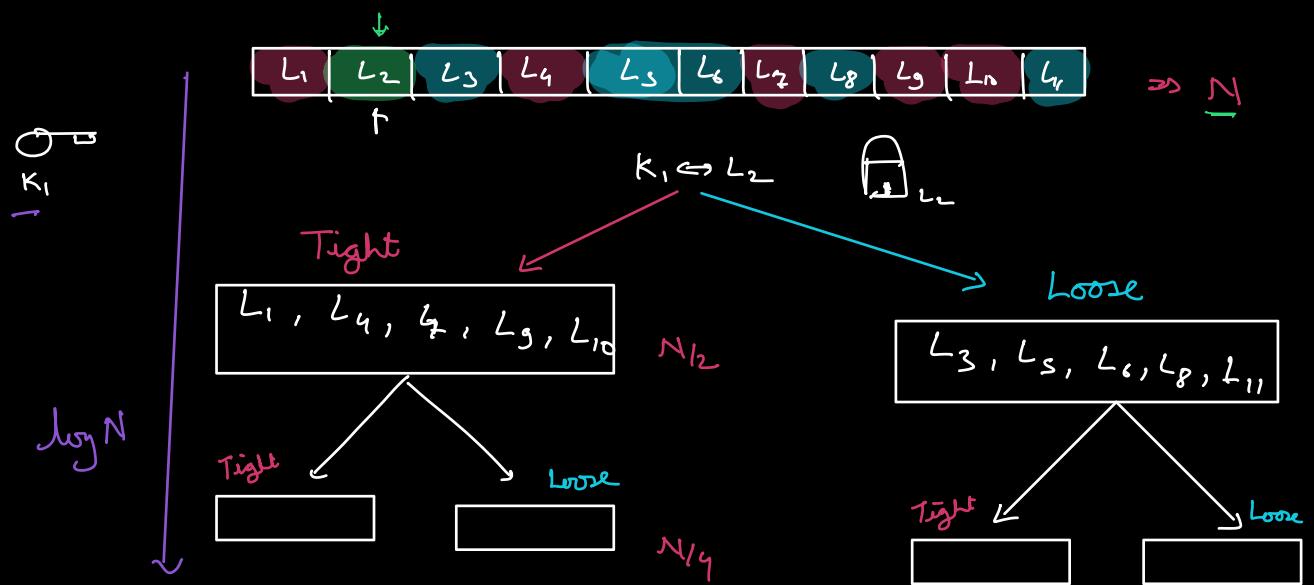
Q Given 1 bag full of locks & 1 bag full of keys. Every lock has (distinct) one key in the key bag. ( $\perp - \rightarrow - \perp$  mapping)

Comparing a key with other key

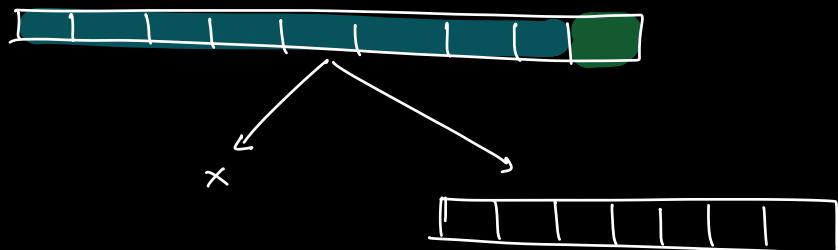
& a lock with other lock is not flexible.

Find the matching key for every lock.





$T_C : O(N \log N) \leftarrow B_C$   
 $\underline{O(N^2)} \leftarrow W_C$



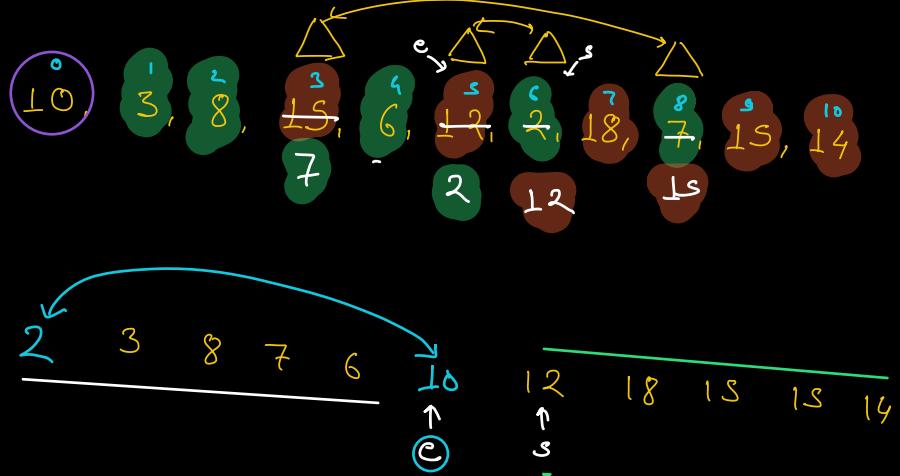
Bubble  
 Merge ~~X~~  
 Selection

Q Given an array of N elements. Rearrange the array such that :

All elements  $\leq A[0]$  go to the left of it

All elements  $> A[0]$  go to the right of it

$A : \textcircled{10} \ 3, 8, 15, 6, 12, 2, 18, 7, 15, 14$   
 $\Rightarrow 3, 8, 6, 2, 7, 10, 15, 12, 18, 15, 14$



$\text{swap}(A[0], A[s]) \times$   
 $\text{swap}(A[0], A[s-1]) \checkmark$   
 $\text{swap}(A[0], A[e]) \checkmark$

$A : \textcircled{7} \ 4 \ 2 \ 3 \ 4 \ 5 \ 8$   
 $\textcircled{e} \ s$

$\text{while } (s \leq e)$   
 $\quad \quad \quad \left\{ \begin{array}{l} \text{if } (A[0] \geq A[s]) \\ \quad \quad \quad s+1 \end{array} \right.$   
 $\quad \quad \quad \text{if } (A[0] < A[e]) \\ \quad \quad \quad e--;$   
 $\quad \quad \quad \text{swap}(A[s], A[e])$   
 $\quad \quad \quad s+1; e--;$

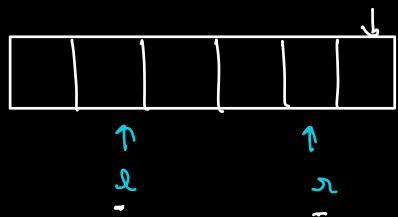


Q Given an array. Re-arrange the sub-array from  $l$  to  $r$ . Such that

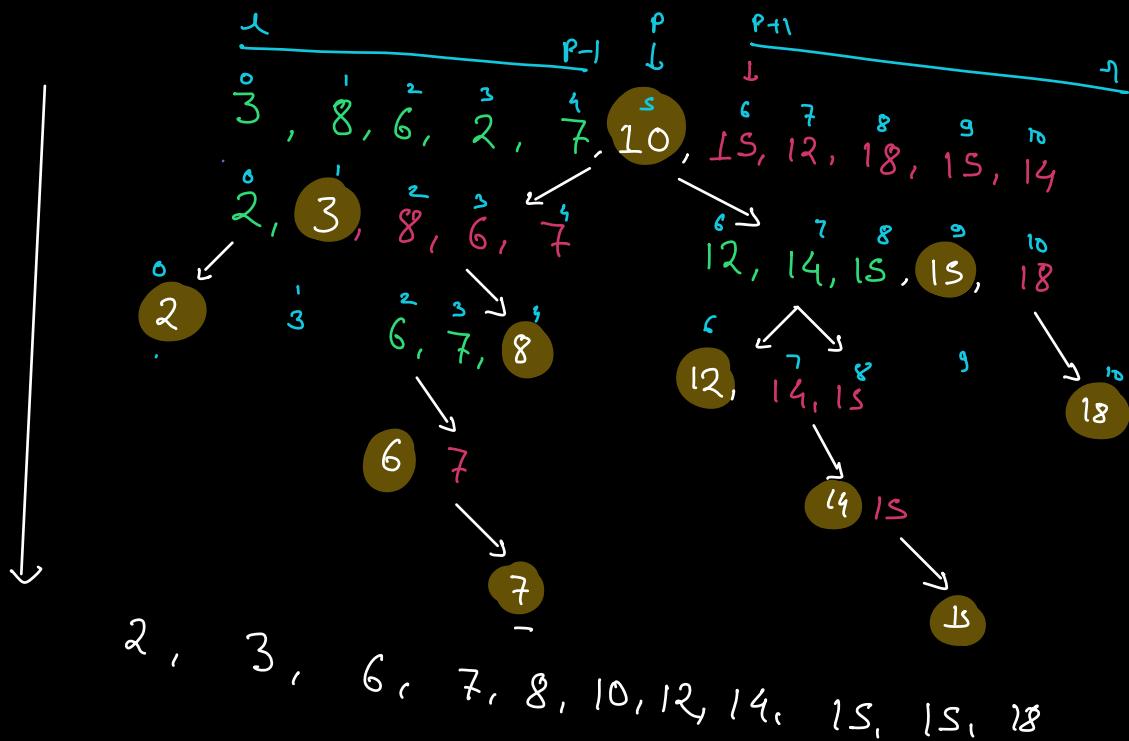
All elements  $\leq A[l]$  go to the left of  $il$   
 $(\in [s, e])$

All elements  $> A[l]$  go to the right of  $il$   
 $(\in [s, e])$

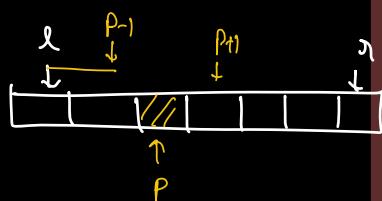
```
int Partition (A, l, r) {
    s = l+1; e = r;
    while (s <= e) {
        if (A[l] >= A[s]) {
            s++;
        } else if (A[e] > A[l]) {
            e--;
        } else {
            Swap(A[e], A[s]);
            e--;
            s++;
        }
    }
    Swap(A[l], A[s-1]);
    ret s-1;
}
```



$A : \begin{array}{c} 10 \\ 3, 8, 15, 6, 12, 2, 18, 7, 15, 14 \\ 3, 8, 6, 2, 7, 10, 15, 12, 18, 15, 14 \end{array}$   
 Sorted  $A : 2, 3, 6, 7, 8, 10, 12, 14, 15, 15, 18$



## Quick Sort

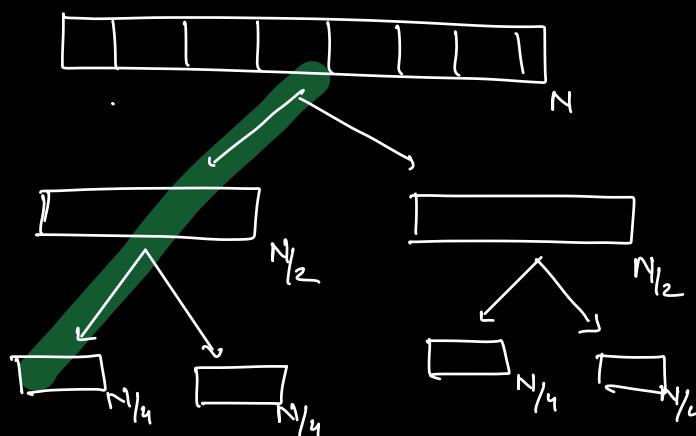


```

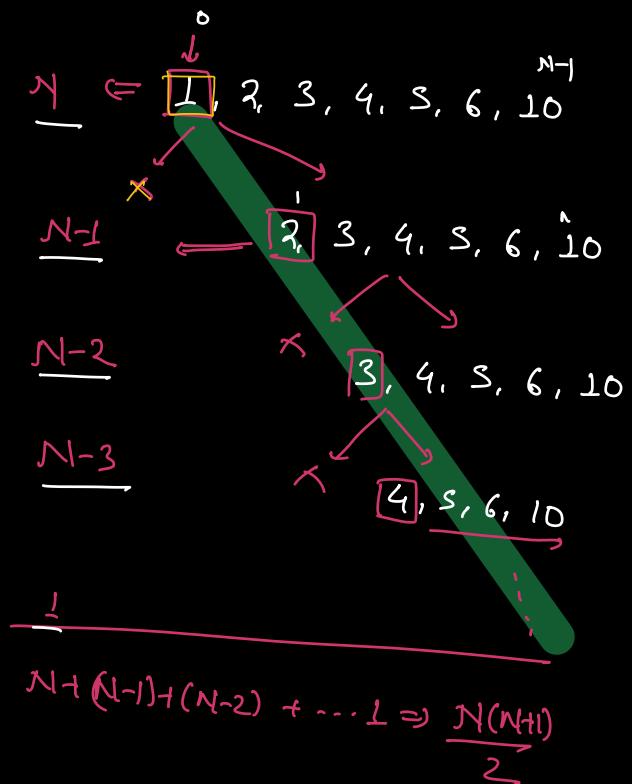
void QuickSort(A[], l, r) {
    if (l >= r) return;
    // Assumptions: QuickSort(A, l, r) → Sort the
    // subarray from l to r,
    P = Partition(A, l, r); => O(N)
    QuickSort(A, l, P-1); → T(N/2) BC
    QuickSort(A, P+1, r); → T(N/2) BC
}

```

TC : Best Case :  $O(N \log N)$   
 Worst Case :  $O(N^2)$   
 \* Avg Case :  $O(N \log N)$   
 Stable : No  
 Inplace : Yes



SC : Best Case :  $O(\log N)$   
 Worst Case :  $O(N)$   
 Avg Case :  $O(\log N)$



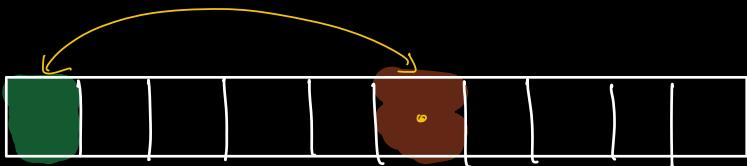
If we pick the pivot randomly then  $O(N^2)$

$O(N^2)$  becomes practically impossible

Probability of picking smallest element every time

$$= \frac{1}{N} \times \frac{1}{N-1} \times \frac{1}{N-2} \times \frac{1}{N-3} \times \dots \times 1$$

$$\frac{1}{N} \times \frac{1}{N-1} \times \frac{1}{N-2} \times \frac{1}{N-3} \times \dots \times 1 \Rightarrow \text{So small that it is practically impossible}$$



Randomized Quick Sort :

$$\text{RandInd} = \text{rand}(1, n) \rightarrow [0, n]$$

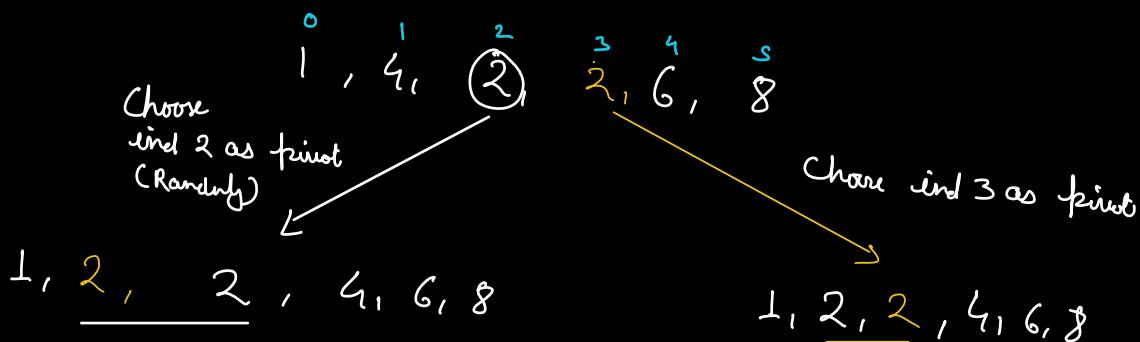
$\text{Swap}(A[l], A[\text{RandInd}])$ ,

$P = \text{Partition}(A, l, r)$ ,

$QS(A, l, p-1)$ ,

$QS(A, p+1, r)$ .

Stability



$\Rightarrow$  Not stable !

	Best Case	Worst Case	Avg Case
Selection Sort	$O(N^2)$	$O(N^2)$	$O(N^2)$
Bubble Sort	$O(N)$	$O(N^2)$	$O(N^2)$
Merge Sort	$O(N \log N)$	$O(N \log N)$	$O(N \log N)$
Insertion Sort	$O(N)$	$O(N^2)$	$O(N^2)$
Quick Sort	$O(N \log N)$	$O(N^2)$	$O(N \log N)$

A : 170, 45, 75, 90, 802, 24, 2, 66 ↗

Based on 0<sup>th</sup> digit (LSD) : 170, 90, 802, 02, 24, 45, 75, 66  $\Rightarrow O(N)$

Based on 1<sup>st</sup> digit : 802, 002, 024, 045, 066, 170, 075, 090  $\Rightarrow O(N)$

Based on MSD : 2, 24, 45, 66, 75, 90, 170, 802  $\Rightarrow O(N)$

Complete array  
→ Sorted order.

HashMap  
 $\langle$ Int, List $\rangle$

$0 \rightarrow [170, 90]$   
 1  
 $2 \rightarrow [802, 2]$   
 3  
 $4 \rightarrow [24]$   
 $5 \rightarrow [45, 75]$   
 $6 \rightarrow [66]$   
 8  
 9

Radix Sort

TC ;  $O(dN)$

Count Sort

c a c b d a b

int [26]

HashMap.    c : 2  
              a : 2  
              b : 2  
              d : 1

a → z

a a b b c c d