HW-2 - Sorting Tracing

(1) **Insertion sort**:- A simple comparison based sorting algorithm. It inserts every array element into its proper position.

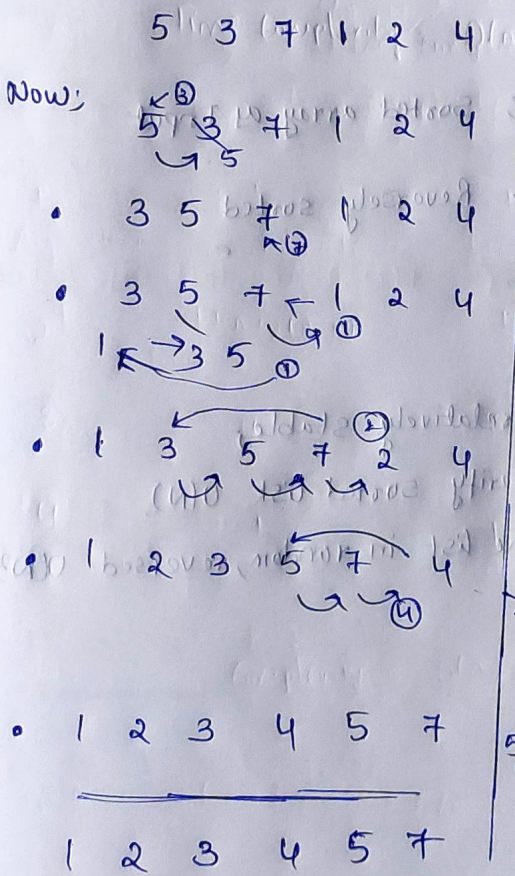
algorithm:-

1. 1st element, already sorted then return 1;
2. Next element.
3. Compare with all elements in the sorted sub list.
4. Shift all the elements in the sorted sub-list that is greater than the value to be sorted.
5. Insert the value.
6. Repeat until the list is sorted.

Tracing:- let us take;

5 3 7 1 2 4

Now;



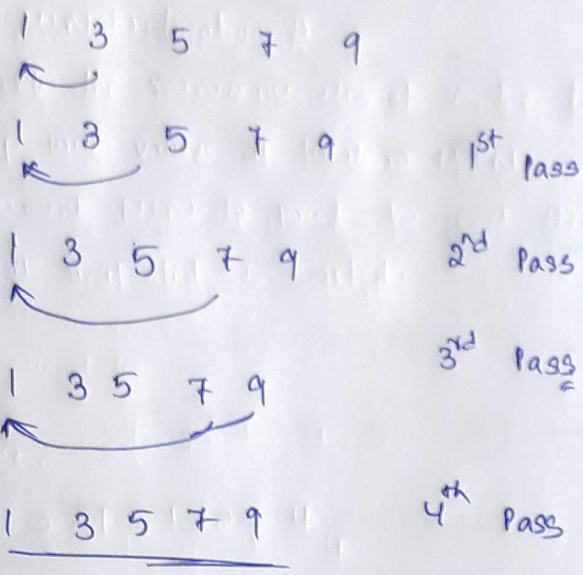
- 3 5 7 1 2 4 — 1st Pass :- Save=1 ; C=2 ; Shift=1 ; Insert=1 ; Total=5
- 3 5 7 1 2 4 — 2nd Pass :- Save=1 ; C=1 ; Shift=0 ; Insert=1 ; Total=3
- 1 3 5 7 2 4 — 3rd Pass :- Save=1 ; C=4 ; Shift=3 ; Insert=1 ; Total=9
- 1 2 3 5 7 4 — 4th Pass :- Save=1 ; C=4 ; Shift=3 ; Insert=1 ; Total=9
- 1 2 3 4 5 7 — 5th Pass :- Save=1 ; C=3 , Shift=2 ; Insert=1 ; Total=7

1 2 3 4 5 7

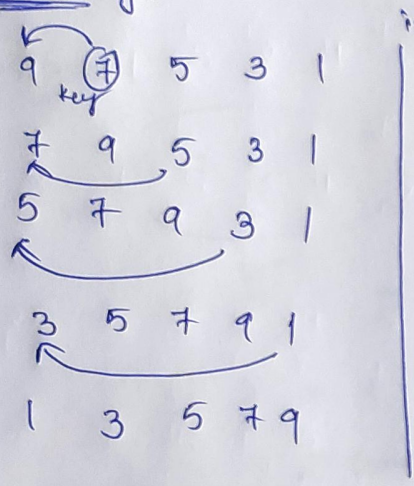
Total Steps = 33

Tracing of sorted array:

let us take;



Reversed array:



index.	key
7	9
5	7
3	.
1	.
-	.
	(n-1)

Best case = $O(n)$ → $1+2+3+\dots+(n-2)+(n-1)$

Total pairs = $\frac{(n-1)}{2}$ → $(1+(n-1))+(2+(n-2))\dots$

$O(n)$

Worst case = $O(n^2)$

→ $n+(n-1)+\dots+1 = \frac{n(n-1)}{2}$

$= O(n^2)$

Big O :-

for $N = 6$; $O(N^2) = 36$ steps

Best case :-

elements already in order

Steps = 15 (close to 6)

Run time : $O(N)$

Worst case :-

elements in reverse order

Number of steps = 52 (close to 36)

Run time = $O(N^2)$.

Time Complexity :-

- Best Case :- $O(N)$; Sorted array as Input
- Worst Case :- $O(N^2)$; Reversely sorted

Finally,

Insertion sort is relatively stable,

- best used to verify sorted list $O(N)$.
- worst when the list in random, reversed $O(N^2)$

2.) Selection Sort :-

Algorithm :-

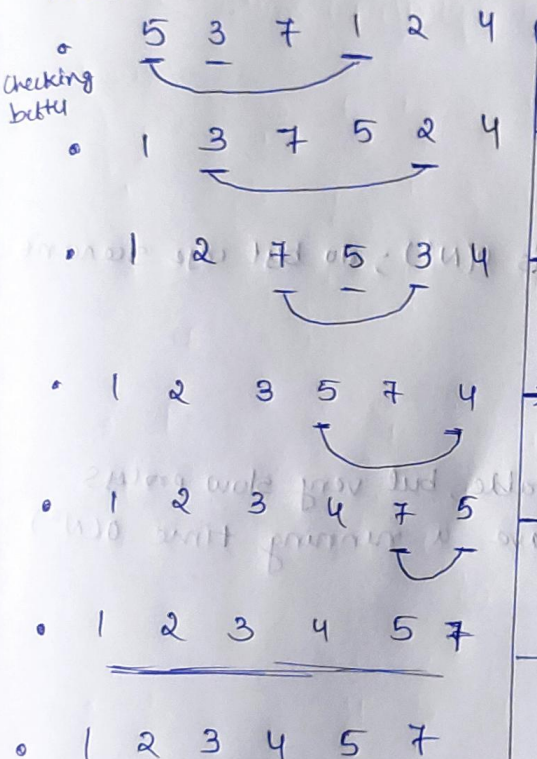
1. Set MIN to location 0.
2. Search the minimum element in the list
3. Swap with value at location MIN.
4. Increment MIN to point to next element
5. Repeat until list is Sorted.

Tracing :-

let us take;

5 3 7 1 2 4

Now;



Counting

→ I^st pass :- C=5 ; B=3 ; S=3 ;
Total = 11

→ 2^nd Pass :- C=4 ; B=2 ; S=3
Total = 9

→ 3^rd Pass :- C=3 ; B=3 , S=3
Total = 9

→ 4^th Pass :- C=2 ; B=2 ; S=3
Total = 7

→ 5^th Pass :- C=1 ; B=2 ; S=3
Total = 6

Total steps = 42

Tracing of sorted array :-

let us take;

① 3 5 7 9

1 ③ 5 7 9 ⟶ 1st Pass

1 3 ⑤ 7 9 ⟶ 2nd Pass

1 3 5 ⑦ 9 ⟶ 3rd Pass

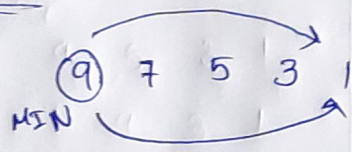
1 3 5 7 ⑨ ⟶ 4th Pass.

Complexity → $(n-1)+(n-2)+\cdots+1$

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

$= O(n^2)$

Reverse Sorted :-



$C = (n-1)$

1 7 5 3 9 $C = (n-2)$

1 3 5 7 9

2nd comparison,

1 3 5 7 9 1st comparison

Complexity :- $(n-1)+(n-2)+\cdots+2+1$

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

$= n^2 + \ldots$

Best & worst case = $O(n^2)$

Big O ;

for N = 6 : $O(N^2) = 36$ steps

Closet.

Best case :-

elements already in order.

Steps = 35 (close to 36)

Run Time = $O(N^2)$

Worst case :-

Elements in reverse order.

Steps :- 52 (close to 36)

Run time = $O(N^2)$

Time Complexity :-

- Best case = $O(N^2)$
- Worst Case = $O(N^2)$

* Time complexity in all cases is $O(N^2)$; no best case scenario

Finally ;

selection sort is very stable, but very slow process. and will always have a running time $O(N^2)$