27/4/2021

DA & A

Sorting Algorithms :

why Sorting :

→ There are obvious applications

i.e, phone book, mp3 orgnaizers

Spreadsheet, documents orgnaizers.

→ Problems becomes easy once items are sorted.

ex:

finding a median :

array A [0 : n]

it contains n numbers and they are not sorted.

After sorty

B [0 : n]

in ascending or descending orders.

All you need is a comparison function through which you can find which item is greater and which is smaller.

Bottom line:

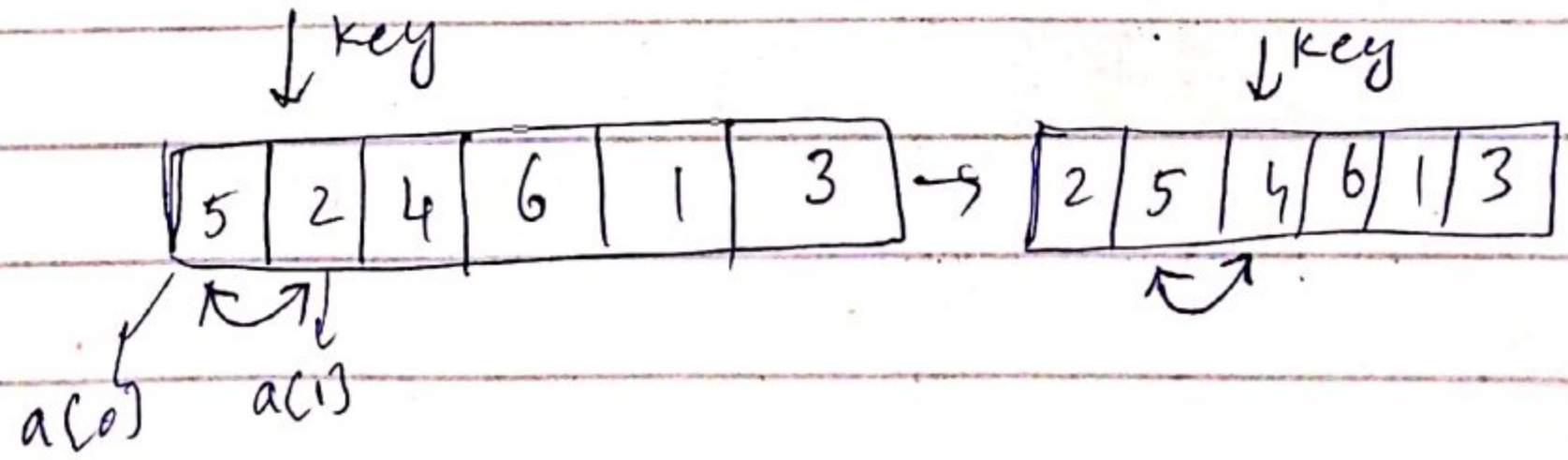
If you have a comparison function, you are going to get sorted elements after some amount of time.

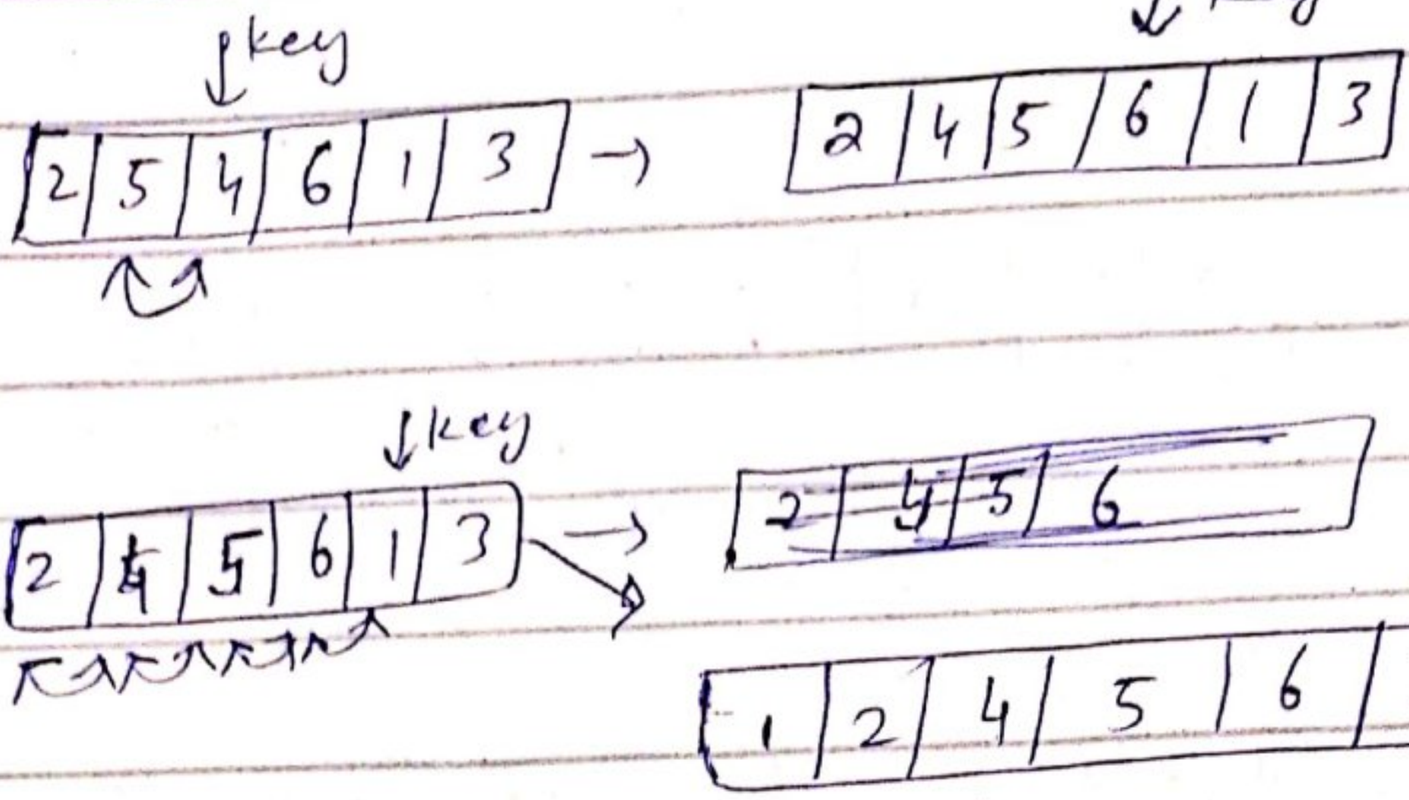
Insertion Sort:

for $i = 1, 2, \ldots n$

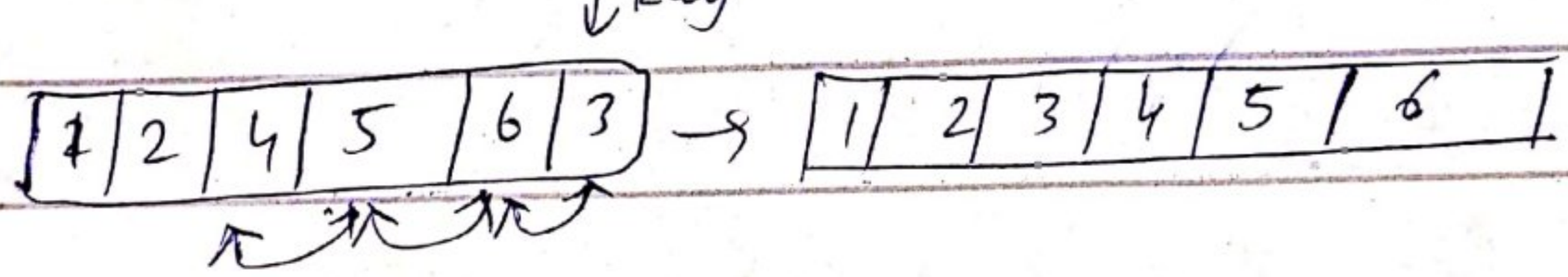
insert $A[i]$ into sorted array $A[0 : i-1]$

by pairwise swaps down to the correct position



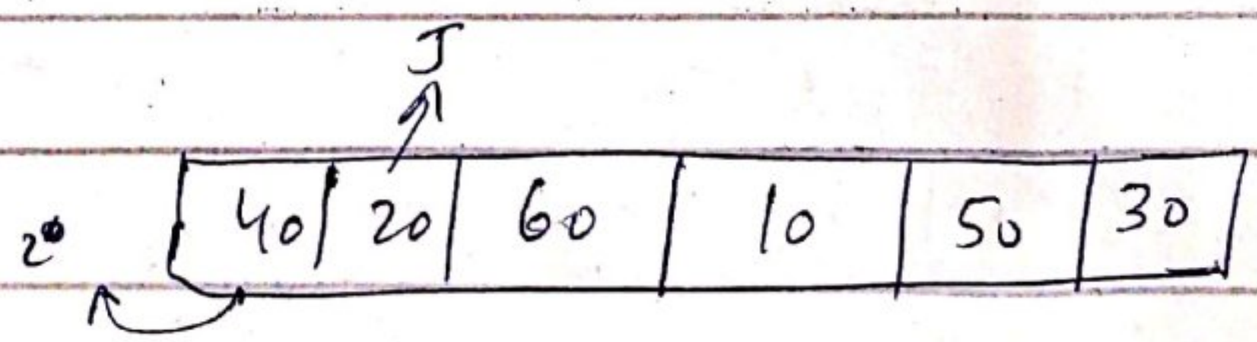


These four swaps takes place in different four operations / steps as we are following pairwise swapping operation.



Ex-2

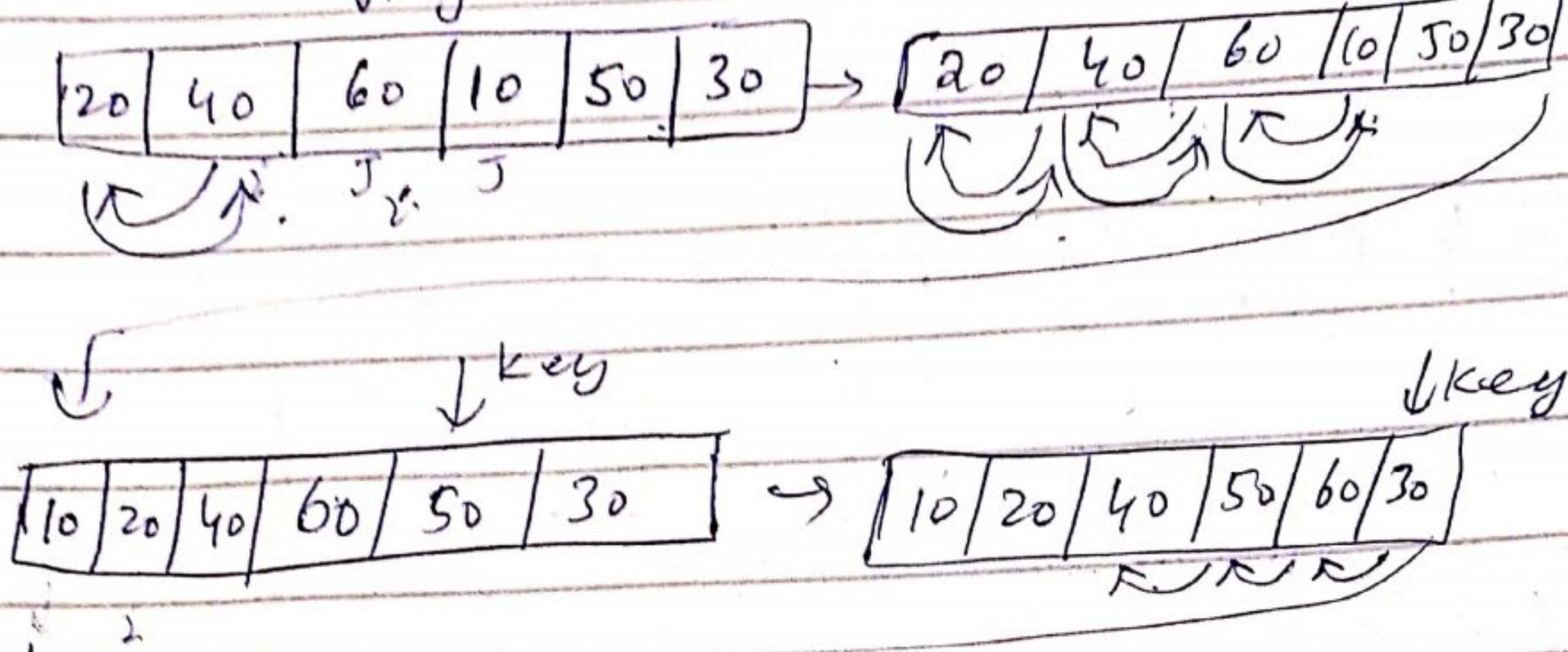
It is one of the major sorting algorithm.



→ first element to enter is 40.

→ The next element which is to be inserted is 20 also called key.

20



Psuedo code:

for J = 2 to A-length

key = A[J]

i° = J - 1

while i° ≥ 0 and A[i°] > key

A[i° + 1] = A[i°]

i° = i° - 1

A[i° + 1] = key

Time complexity:

Best Case (Ascending order).

10 20 30 40 50 60

comparison	swaps.
0	0
1	0
1	0
1	0
1	0
(n-1)	O(1)

O(n) → Best case.

Worst case : (descending order).

60 50 40 30 20 10.

compare	swap.	0, 1, 2, 3, --- (n-1)
0	0	↓
1	1	$\frac{n(n-1)}{2} = \frac{n^2-n}{2}$
2	2	
3	3	$O(n^2)$
⋮	⋮	
(n-1)	(n-1)	

Average case:

$O(n^2)$. comparisons and swaps.

Bubble Sort :

10	9	11	6	15	2.

→ Starting from the very first element, and start comparing, if the left side element is bigger then, compare these two, and if the right is biger then shift the bubble to the biger side.

pass-1

9 (10) 11 6 15 2.

9 10 (11) 6 15 2.

9 10 6 (11) 15 2.

9 10 6 11 (15) 2.

9 10 6 11 2 15

after every pass one element will be its correct position in terms of order.

(9) 10 6 11 2 | 15

pass-2

9 (10) 6 11 2 | 15

9 6 (10) 11 2 | 15

9 6 10 (11) 2 | 15

9 6 10 2 | 11 15

pass-3

(9) 6 10 2 | 11 15

6 (9) 10 2

6 9 (10) 2

6 9 2 | 10

pass-4

(6) 9 2 10 | 11 15

6 (9) 2 10 | 11 15

6 2 | 9 10 | 11 15

(6) 2 9 10 11 15

2 6 9 10 11 15

All sorted.

Question will be asked?

→ count total number of swapping.

→ time complexity.

worst Case: (maximum time).

9 6 7 5 4

to bring to it's original position

there will 4 comparisons.

9 6 7 5 4 → 4

(n-1) ; n=5.

n-2

n-3

:

1

So. we can write this series as

$$\frac{n(n-1)}{2} \Rightarrow \frac{n^2 - n}{2}$$

$$O(n^2)$$

cuz we take upper bound in worst case scenario.

S = 1 2 3 4 — n-2 n-1

S = n-1 n-2 n-3, —— 2 1

2S = n n n —— n

= (n-1)n/2