CS & IT ENGINEERING

Algorithms

Sorting Algorithms

Lecture No.- 01

By- Aditya sir

Recap of Previous Lecture

Topic

Topic

TC analysis

Topics to be Covered

- Topic
- Topic
- Topic

Sorting

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored 12,000+ students & working professions in field of Data Science and Analytics
11. Have been mentoring & teaching GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

Telegram Link for Aditya Jain sir:

https://t.me/AdityaSir_PW

Sorting

- Imp Terminologies
- Bubble Sort

Topic : Sorting Algorithms

Sorting

A process of ordering/arranging the given element in a particular sequence/ order as per the given criteria.

Topic : Sorting Algorithms

Eg.1. 50, 79, 3005, 452, 297

#Q. Arrange the given element in increasing order of their values?

→ 50, 79, 297, 452, 3005

1 2 3 4 5

Topic : Sorting Algorithms

Eg.2. 50, 79, 3005, 452, 297

Arrange in Decreasing order of values.

→ 3005, 452, 297, 79, 50,

Topic : Sorting Algorithms

Eg.3. Arrange the given elements in increasing order of their 1st digit.

50,	79,	3005,	452,	279
↓	↓	↓	↓	↓
297,	3005,	452,	50,	79

Topic : Sorting Algorithms

Sorting Algorithms:

1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Merge Sort
5. Quick Sort
6. Heap Sort
7. Radix Sort

DnC

Topic : Sorting Algorithms

For every sorting Algorithms:

Example

Code

Internal working

Complexity analysis

- Time
- Space

Properties

Topic : Sorting Algorithms

Terminologies & Classification:

Sorting Algorithms

Internal Algorithms

Comparison Based
- Bubble sort
- Selection sort
- Insertion sort

Non- Comparison Based
- Radix Sort

External Algorithms
- Merge Sort
- External Insertion Sort

Topic : Sorting Algorithms

Comparison–Based Sorting:-

- When an element is compared with the other element of the given data to achieve sorting.

Topic : Sorting Algorithms

Non-Comparison–Based Sorting:-

- When sorting is achieved without any comparison among the elements to be sorted.

eg:- Radix Sort

Topic : Sorting Algorithms

Other classification of sorting Algorithms:-

1. Internal vs External
2. Comparison vs non-comparison
3. Recursive vs Iterative
4. Inplace vs not-inplace
5. Stable vs unstable.

V. Imp

Note:- If nothing is specified, the default sorting order will be as ending/increasing.

Topic : Sorting Algorithms

In-place sorting Algorithm:-

Additional/Auxiliary space Requirement

- At most O(1) {excluding recursion stack}
- Recursion stack of atmost O(logn)

Eg.4. Merge Sort:- Space complexity ⇒ O (n + logn)

Topic : Sorting Algorithms

Stable sorting Algorithms:-

- If relative position of two equal elements is maintained before & after sorting.

Topic : Sorting Algorithms

Formal Definition:-

If A[i] = A[j]

&

In given input: A[i] is before A[j]

Then, algorithm is stable if A[i] is before A[j] in finally sorted (after sorting) array as well.

Topic : Sorting Algorithms

Eg.5. Stable vs unstable:

Topic : Sorting Algorithms

Inversion in an Array

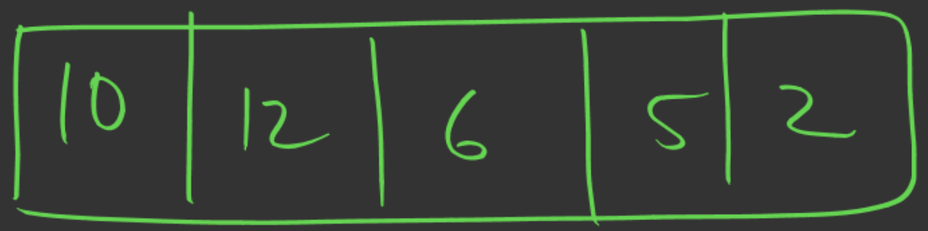
Formal Definition:

- A pair of indices (i,j) is considered to be an inversion of the given Array A[1...n] if i < j and A[i] > A [j]

Note:-i & j need not be adjacent indices (not mandatory)

In simple terms, if a larger elements is present before a smaller element, then their indices will form an inversion pair.

A:



Topic : Sorting Algorithms

Eg.6.

	1	2	3	4	5
A:	10	12	6	5	2

Total number of inversion pairs in given array A?

(1, 3) (1,4) (1,5) → 3

(2, 3) (2,4) (2,5) → 3

(3, 4) (3,5) → 2

(4, 5) → 1

9

Total number of inversion = 3+3+2+1 = 9

$A[n]$

Topic : Sorting Algorithms

Range of number of inversion in given array:

Given A [1.....n] max min

Min number of inversions

→ A is in increasing order

	1	2	3	4
A:	5	10	15	18
Inv:	0	0	0	0

N = 4

Total inversion = 0

Topic : Sorting Algorithms

Range of number of inversion in given array:

Given A [1.....n] max min

2. Max number of inversions

→ A is in decreasing order

	1	2	3	4
A:	18	15	10	5

(n-1)

N-elem. (n-2)

(n-3)

In general

$$\left(\sum_{i=1}^{n} i\right) - n$$

$$(n-1) + (n-2) \ldots\ldots + 0 = \frac{n(n+1)}{2} - n$$

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

Topic : Sorting Algorithms

$$A \rightarrow n$$

$$\boxed{0 \leq |\text{numberofinversion}| \leq \frac{n(n-1)}{2}}$$

$$\left(\sum_{i=1}^{n} i\right) - n = \frac{n(n+1)}{2} - n = \frac{n^2 + n - 2n}{2} = \frac{n^2 - n}{2} = \frac{n(n-1)}{2}$$

Topic : Sorting Algorithms

Time complexity of any comparison based sorting algorithms:-

- depends on two important factors:
  1. Number of comparison:
  2. Number of Swaps (number of inversion).

Topic : Sorting Algorithms

Bubble sort→ by default: ascending order

→ Basic Algorithm .....(Algorithm -1)

→ Better Algorithm .....(Algorithm -2)

→ Optimised Bubble sort .....(Algorithm -3)

Logic/ Idea:- In every i^{th} pass/ iteration, place the i^{th} max element at its correct position,

Topic : Sorting Algorithms

Eg.6. Given

	1	2	3	4	5	6
A:	75	55	15	10	35	5

Given array is input to pass 1

Topic : Sorting Algorithms

Pass -1	75	55	15	10	35	5
After Pass -1 →	55	15	10	35	5	75
Pass -2	15	10	35	5	55	75
Pass -3	10	15	5	35	55	75
Pass -4	10	5	15	35	55	75
Pass -5	5	10	15	35	55	75

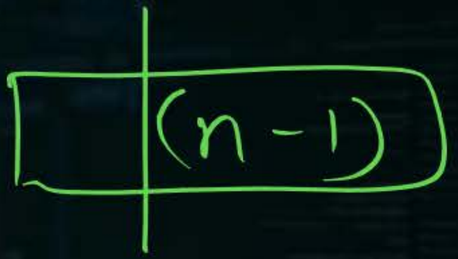
Sorted

Topic : Sorting Algorithms

Imp. Observation

After pass 1 → 1 elements

After pass 2 → 2 elements



After pass $(n-1)^{th}$ → (n-1) elements will be at their correct position.

Note:- The max number of passes required to = (n–1) passes.

Topic : Sorting Algorithms

Basic Bubble sort

```
Algorithm Bubble Sort  (A, n)
{
        for (pass =1 ; Pass <= (n-1) ; Pass ++)
        {
                for (j = 1; j <= (n-1) ; j++)
                {
                        if (A[j] > A[j +1])
                        {
                                Swap (A[j] , A [j +1])
                        }
                }
        }
}
```

1 based indexing

TC analysis

Topic : Sorting Algorithms

1. (n-1)
2. (n-2)

Algorithm time complexity

	No. of comparison	No. of Swaps
(inc.) Best case	$(n-1)*(n-1) = (n-1)^2$	0
(dec.) Worst case	$(n-1)*(n-1) = (n-1)^2$	$\frac{n(n-1)}{2}$ $\Rightarrow (n-1)+(n-2)\ldots 1$

$\Omega(n^2)$

$O(n^2)$

$\Theta(n^2)$

2 mins Summary

- Topic
- Topic
- Topic
- Topic
- Topic

Terminologies

Bubble → Algo

THANK - YOU