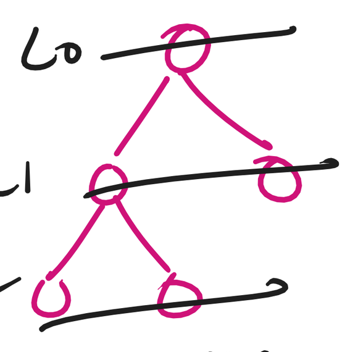
BFS / DFS
↓
LOT

Abdul Bari
GFG

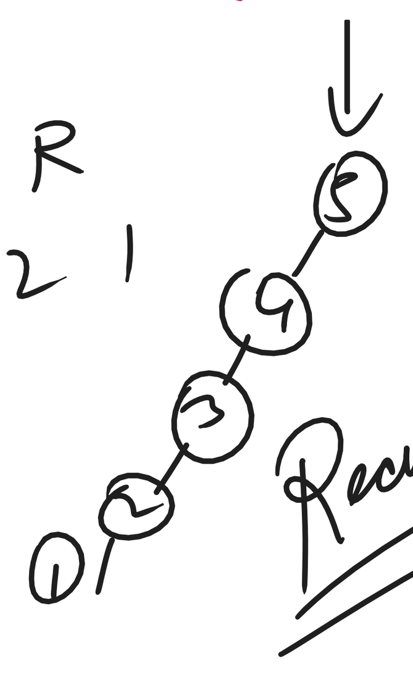
Data Structures & Algorithms



Linear → Arrays, S, Q, LL

Non-linear → Trees [Graphs] Tries

Wipro
↓
L < N < R
5 4 3 2 1



Recursion

- CBT (Heap) → Heap Sort
- Binary Trees / Search Trees → BST, AVL, Red Black Tree
- Skewed Tree
- Orthogonal Range Trees
- Fenwick Trees
- Fenwick
- K-D Trees

Stacks → Reversal

① → ② → ② — ①

T1

T2

Palindrome

1 →
2
2
1

Note:

DS — Appⁿ

Tra

Leet Code → CC

Algorithms: →

[Oracle] pivot [ADA]

Searching
- * Linear Search → O(n)
- * Binary Search → s, e → log n
- * Recursive Binary Search → log n
- * Jump Search → √n
- * Interpolation Search →

Sorting
- * Bubble Sort
- * Selection Sort
- * Insertion Sort

Comparison → $O(n^2)$

- * Merge Sort
- * Quick Sort

→ $O(n \log n)$

Non-comparison Algos
- * ① Count Sort
- ⑪ Radix Sort

Time Complexity:

Count Sort Algorithm: →

- * Single digits (0-9)
- * Non-comparison algo

Steps: →
1. Find the max = 6
2. Create a count array 0 – max / 0 – 6
3. Calculate the freq of each element
4. Cumulative count
5. o/p array
6. Start from the end

$O(n+max)$

prev | S1 | S2 | S3 |
| 1 | 2 | 3 | 9or 10

S1
0 0 0 1 1 1 2
1 2 3 4 5 6 7th

i/p

6	4	3	4	1	3	2	3	1
0	1	2	3	4	5	6	7	8

→ len

0	2	1	3	2	0	1
0	1	2	3	4	5	6

→ count array

0 → 1 → 2 → 5 → 4 → 3 → 7 → 6 → 8

L→R

0	2	3	6	8	8	9
0	1	2	3	4	5	6

→ Cumulative count
c = c + p

1	1	2	3	3	3	4	4	6
0	①	2	3	4	5	6	7	8

→ Sorted Array

* Radix Sort Algorithm: →

- * Non-comparison Algorithm
- * Multi-digit Numbers
- * Constant Length Strings

"pavan" "raman" "sameer"

→ Unit's, ten's, hundred's ...

n ele
max

$O(n+max)$

count sort

	325	009	246	082	001
P1	001	082	325	246	009
P2	001	009	325	246	082
P3	001	009	082	246	325

* Bucket Sort

S→1 Create (0 – 9) 10 buckets

S→2 Find the max → 325

no of digits → 3 → 3 passes

0	1	2	3	4	5	6	7	8	9
	001	082			325	246			009
001, 009		325		246				082	
001, 009, 082		246	325						

Important points about Radix Sort: →

* Each iteration of radix sort focuses on single digits. Therefore, in each iteration we call the count sort method to sort the single elements.

* Also, the time complexity is similar because n elements are sorted & we find the max value for the no. of iterations. Big O (n + max).

* Therefore Radix & Count Sorts are called siblings.

* How is the number of iterations / steps / passes controlled in Radix Sort Algorithm ?

{Infosys | Wipro | Capgemini | IBM | HCL | TCS}

```
for (int exp = 1; max/exp > 0; exp *= 10){
    countSort (arr, exp);
}
```

1, 2, 3

exp = 1000

max = 325

325 / 1 = 325
325 / 10 = 32
325 / 100 = 3
325 / 1000 = 0