Hamming Weight ?

count = 0
++ = 1
++ = 2
++ = 3
↓
return

```
int hammingWeight(int n) {
```

11 → 1011
& 0001
0001 → True

101 >> 1 = 101
001
001 → T

10 >> 1 = 10
01
00 → F

10 >> 1 = 1
1
1 → T

1 >> 1
→ 0 → stop

Terminating coding Questions!

}

Reverse integer :→ n = 123 o/p 321

```
int reverseInteger (int n) {
ans = 0
[ans = ans×10 + digit]
}
```

n = 123

[digit = n % 10
123 % 10 = 3]

(n/10)
123/10 = 12
12/10 = 1

n % 10
12 % 10 = 2
1 % 10 = 1
1/10 = 0
stop

= 0 + 3
= 3
3×10 + 2 = 32

123 = 1×100 + 2×10 + 3×1
32×10 + 1 = 321

(153) = $1^3 + 5^3 + 3^3$
= 1 + 125 + 27 = 153
3

(w)(x)(y)(z)
$w^4 + x^4 + y^4 + z^3$
= wxyz

Given an array of only 0's 1's & 2's sort the array without using any sorting algorithm.

$C_0 = 0$ 1

int arr[] = {2, 1, 0, 1, 2, 1, 0}

GFG
TCS | Capgemini

o/p = {0, 0, 1, 1, 1, 2, 2}

Time Complexity → O(n)
Space Complexity → O(1)

c0 = 2-1-1=0
c1 = 3-1-1-1=0
c2 = 2-1-1

0 0 1 1 1 2 2

i=0 i=1 i=2 i=3 i=4 i=5 i=6
i/p → 2 0 1 2 0 0 1

2 1 1 1 2 1
0

0 ← C0 → 1+1+1 = 3-1-1-1=0 stop
✓ C1 → 1+1 = 2
✓ C2 → 1+1 = 2

SC (O(1))

O(n)

```
index = 0
while ((c0 > 0)) {
    arr[index++] = 0;
    c0--;
}
```

[0 0 0 2 2 1 1]
0

O(n) while O(n)
while O(n)
while O(n)

Data Structures & Algorithms :→ (Roadmap)

Algorithms :→ Searching & Sorting Algos

Kadane Algo?

Searching Algos:

1. Linear → O(n)
2. Binary → Log N
3. Recursive Binary → Log N
4. Jump Search
5. Interpolation Search

$O(n^2)$

Sorting : Skills

- Bubble →
- Selection → Swap
- Insertion → Shift
- Merge → n log n
- Quick → n log n O(1)

NCA { Count Sort, Radix Sort]

result = 0

1, 2, 3, 4, 3, 2, 1 (struck through)

S - 0
D - 1

^
0000
0001
0001
0010
0011
0011
0000
0100
0100

0100
0011
0111
0010
0101
0001
0100 = (4)