31 Jan

31 January 2022 21:05

GCD or HCF :-

a	b
27	18

a = 27
b = 18

b) a (1
18) 27 (1
18
9) 18 (2
18
0) 9

```
while(a % b != 0) {
    r = a % b;
    a = b
    b = r
}
return b;
}
```

a = 10
b = 15

10 % 15 = 10 (r)

a = b | a = 15
b = r | b = 10

```
public static int gcdRec(int a, int b) {
  1 int r = a % b;
  2 if (r == 0) return b;
  3 return gcdRec(b, r);
}
```

a	b	r	
8	4	0	✗ 2
12	8	4	✗ ✗ 3
56	12	8	✓ ✗ 3
68	56	12	✓ ✗ 3
124	68	56	✓ ✗ 3
			4

4) 8 (2
8
0

```
public static int gcd(int a, int b) {
  1 while (a % b != 0) {
  2     int r = a % b;
  3     a = b;
  4     b = r;
    }
    return b;
}
```

a = ~~27~~ 18
b = ~~18~~ 9
r = 9

Power :-

$n^x = 2^5 = 32$

$n^1 = n$

$n^x = n \times n^{(x-1)}$

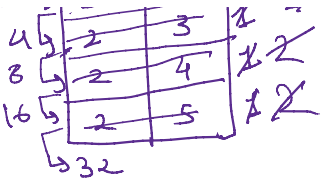
2^5

$2^1 = 2$

n	x	
2	1	1 ✓
2	2	✗ 2
2	3	✗ 2
2	4	✗ 2
		✓

2 → 4 → 8

```
pow(n, x) { if(x==0) ret
1. if (x==1) return n;
2. return n * pow(n, x-1);
}
```



}

$2^{10} = 1024$

$n/2 \quad n/2$

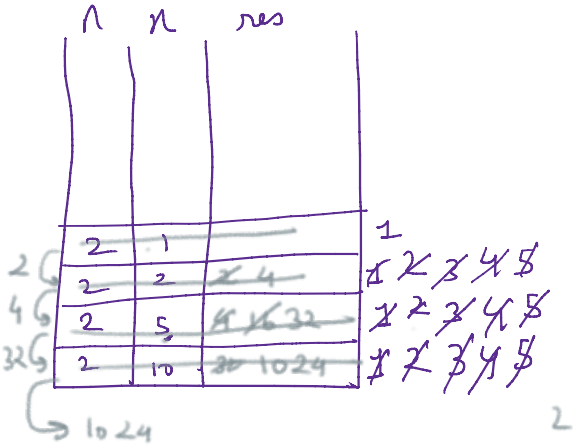
$2^5 \times 2^5 = 2^{10}$

$n/4 \quad 2 \quad n/2$

$2 \times 2 \times 2 = 2^5$

$2^1 \times 2^1 = 2^2$

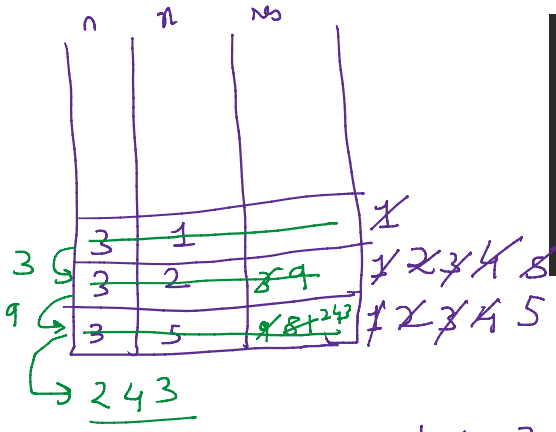
```
powO(n, n)
if(n==1) return n;
    res = powO(n, n/2);
res = res * res;
if(n%2==1){
  res = res * n
}
return res;
}
```




```
public static int powO(int n, int x) {
1  if (x == 1) return n;
2  int res = powO(n, x / 2);
3  res = res * res;
4  if (x % 2 == 1) {
       res = res * n;
   }
5  return res;
}
```

$2/2 = 1$

$2^5 = 2 \times 2 \times 2$




```
public static int powO(int n, int x) {
1  if (x == 1) return n;
2  int res = powO(n, x / 2);
3  res = res * res;
4  if (x % 2 == 1) {
       res = res * n;
   }
5  return res;
}
```

3^5 $\quad n \times n^{(n-1)}$

int a = 2.5 = 2

$\log_2(10) = 3$.....

Pow

2^{10}

2^4

2^8

2^7

2^6

2^5

2^4

Pow O

$\begin{bmatrix} 2^{10} \\ 2^5 \\ 2^2 \\ 2^1 \end{bmatrix}$ 4

$\frac{n}{2} = 1$

$\frac{2}{2}$

$\frac{2}{2}$

$4 \sim 3$

$\log(n)$

$2^{10} = 1024$

$\log(1024) = 10$

2^5
2^4
2^3
2^2
2^1

$n = 10$

$\frac{\log_2(n)}{\log_2(10)}$

$\log_2(1024) = 10$

$[\, n^k = n \;\; \log_n(n) = k \,]$

$2^3 = 8$

$2^4 = 16$

$O\ \log(n)$:-

Time Complexity :-

=) Time taken by your code to get the result.

=) Big O notation ('O') :-

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ..... n]

$i \le 10$

0 ——— and ——— n

$n + n = 2n = O(n)$

$n^3 + n^2 = O(n^2)$

[.....] =) size = n , k

n^k	n	kn
21	17	23

n = for (0 — k)
for (0 — n)

constant = $O(1)$

$2^5 = 2 \times 2^{(n-1)}$

$2^4 = 2 \times 2^{(n-2)}$

$2^{(n-3)}$

$2^4 = 2 \times 2^{(n-3)}$

$2^3 = 2 \times 2^{(n-4)}$

$2^2 = 2 \times 2^{(n-5)}$

$2^1 = 2 \times 2$

$2^{(n-4)} = 2^1$