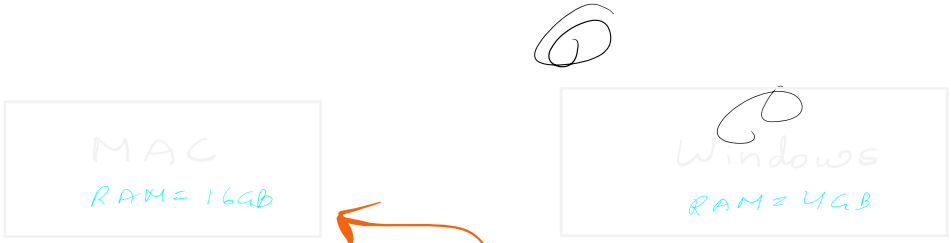
Thursday, September 18, 2025 11:31 PM

Time complexity & Space complexity :-



Program → Odd / Even

?

2s
3s

Big O Notation :-

Input

MAC

1
2
3
4
5
6
7
:
∞

Window

1
2
3
4
5
6
7
:
∞

time / input

sum = n

program → P_1

small ✓

large ✓

very very large ✗

P_1

small ✓

large ✓

very very large ✓

program → time / unit

Rate of increase of input with respect to time

$y = mx$

$t \propto n$

→ It is independent of machine or the programs.

Denote → Big O Notation → $O(n)$ ✓

$n = 5$

O^n :-

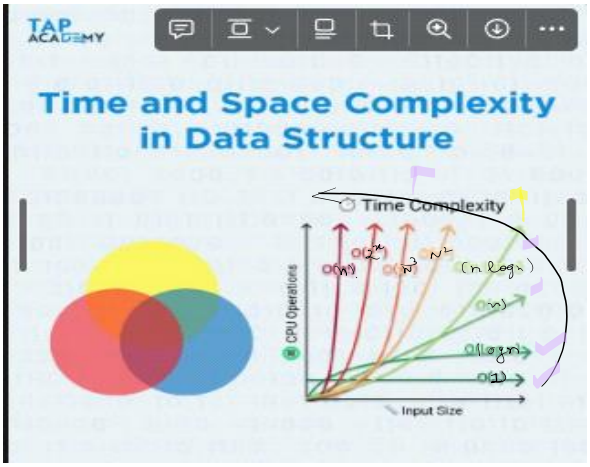
```
for(i=0 → n) {
    print("Name");
}
```

$O(1)$

$i = 5$

$O(N)$ ←

$i < 5$



What is the time, space complexity of following code : [Asked in Amazon]

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
  a = a + rand();
}
for (j = 0; j < M; j++) {
  b = b + rand();
}
```

Assume that rand() is O(1) time, O(1) space function.

O(1) O(1) O(N) O(1) O(M)

total = O(N) + O(M)

⇒ O(M+N)

ignore it

$n^{1000} + n \simeq n^{1000}$

What is the time, space complexity of following code :

```
int a = 0, b = 0;
for (i = 0; i < N; i++) {
  for (j = 0; j < N; j++) {
    a = a + j;
  }
}
for (k = 0; k < N; k++) {
  b = b + k;
}
```

Nested loop:

O(N) × O(N)

$O(N^2)$

```
for (i = 0 → 3) {
  for (j = 0 → 3) {
    print("Hi")
  }
}
```

9 times

Nested → O(N × M) → 2

→ O(N × M × K)

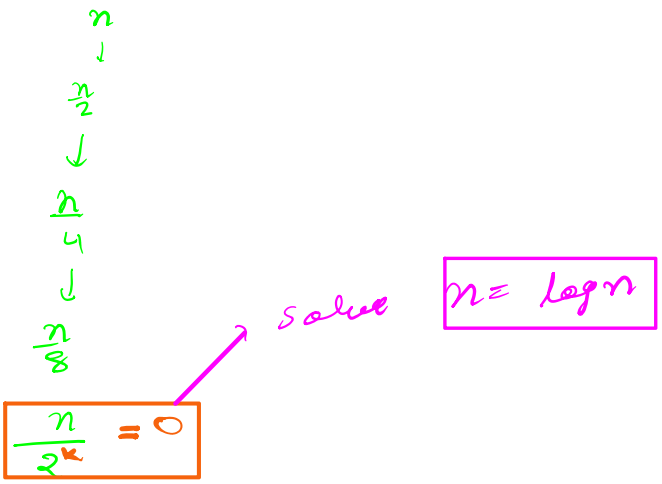
→

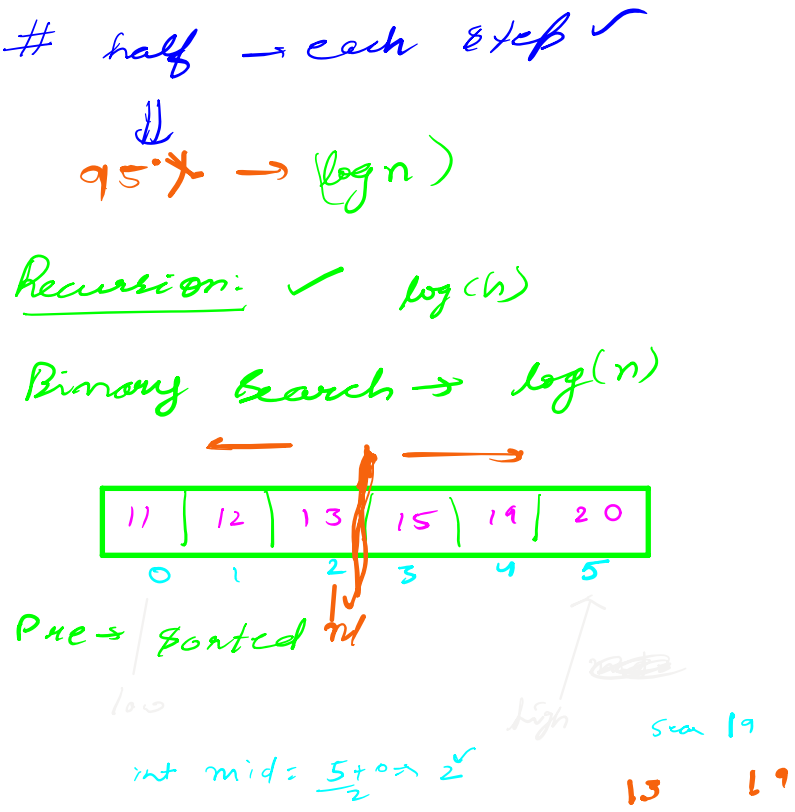
What is the time complexity of the following code :

```
int a = 0;
for (i = 0; i < N; i++) {
    for (j = N; j > i; j--) {
        a = a + i + j;
    }
}
```

What is the time complexity of the following code :

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```





log(n)

What is time complexity of following code : // Asked in Amazon and Microsoft

```
int count = 0;
for (int i = N; i > 0; i /= 2) {
    for (int j = 0; j < i; j++) {
        count += 1;
    }
}
```

O(N)

When i=N, the inner loop runs N times
When i=N/2, the inner loop runs N/2 times
When i=N/4, the inner loop runs N/4 times
And so on

$N + \frac{N}{2} + \frac{N}{4} + \frac{N}{8} + \cdots$ ← this is a geometric expression

So the total number of iteration of inner loop is calculated by

Sum = 2N - 2

time complexity → O(N)

i: N → N/2 → N/4 → ...

$N + \frac{N}{2} + \frac{N}{4} + \cdots$

$(2N - 2) \rightarrow O(N)$

Rule 1 → smaller value / term ✗

Rule 2 → ignore constants ✗

Space Complexity :-

RAM / → space (✗)

O →

space / input

Practical Usage :-

Data → Millions → Billions

Time ↓ → Money → Users

Space ↓ → Cost

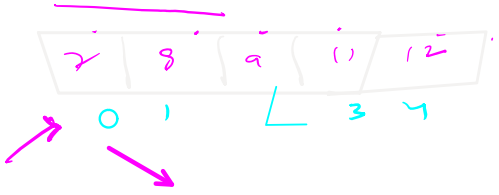
32 RAM — data logo (X)

Worst Case:-

Best Case

Worst Case:

Average Case



Best → 7 → best

Worst → 12 → Worst Case

Worst → 12 → Worst Case

Averg → 9