L34
Binary Search : Introduction

Join Discord - https://bit.ly/ly-discord

RECAP

1. Ferris Wheel Problem
2. Stick Lengths Problem
3. Another problem given as homework (did you try?)

The Secret Number Game

I'll select a number b/w 1 and 10^6 (i.e. 1 mil)

Your target is to ask me some questions and then guess the number finally.

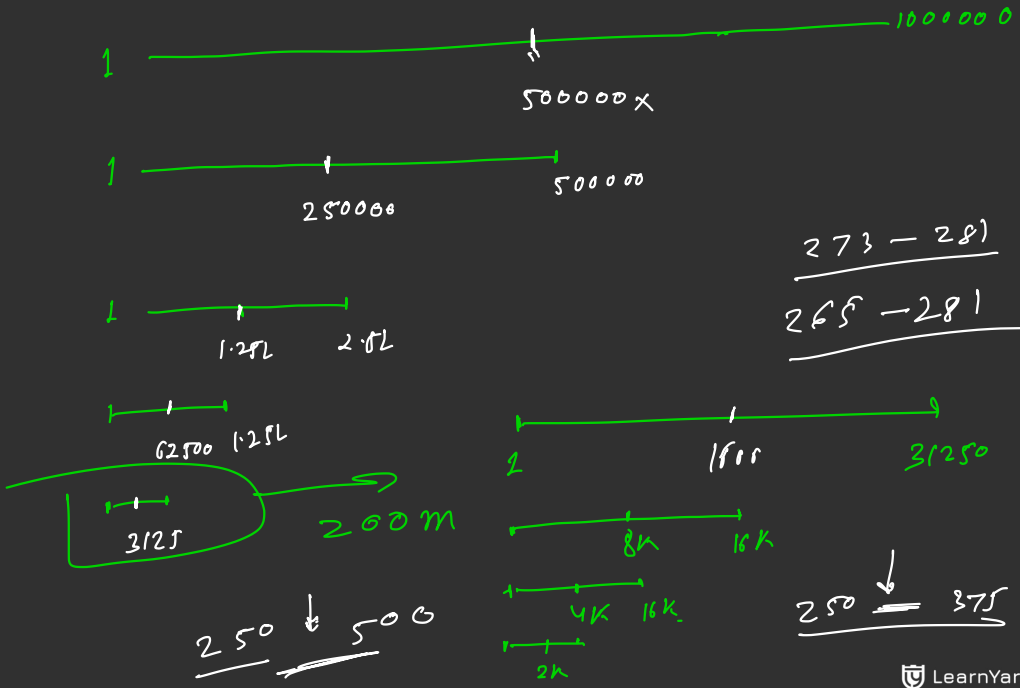
What question can you ask?

In 1 operation, you can choose a number X and ask me:
"Is the secret number greater than X or not?"

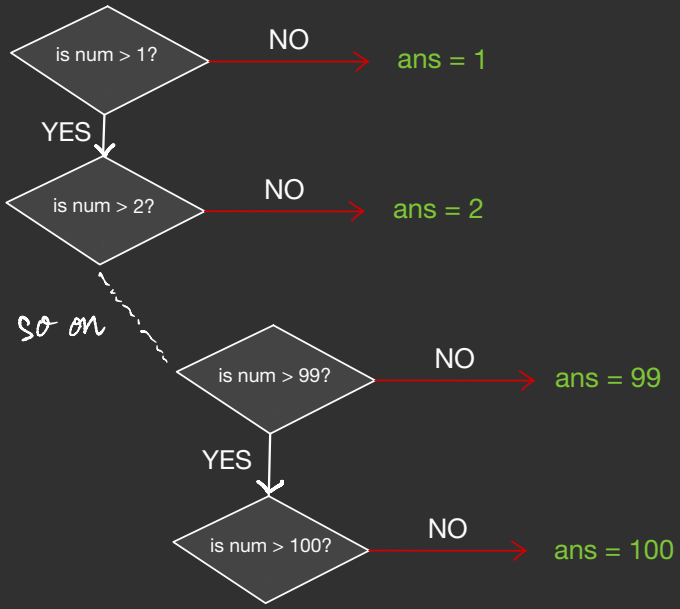
Two rules :

1. I can't change the number once chosen. So, it'll be same always.
2. I won't lie xD

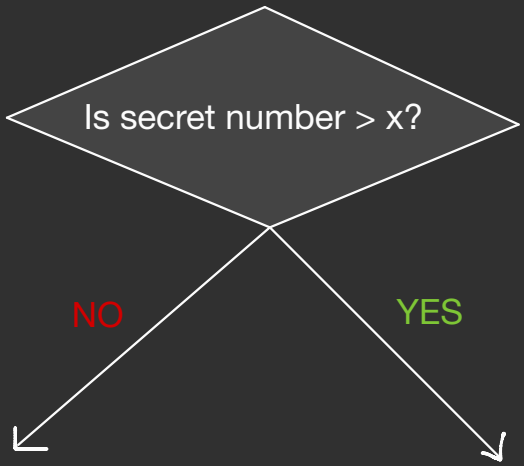
Understanding the strategy for the game


1
1000000
500000
1
500000
250000
273 – 281
265 – 281
1
1.25L
2.5L
62500
1.25L
3125
200m
1
15625
31250
8K
16K
4K
16K
2K
250
500
250
375
LearnYard

Linear Approach

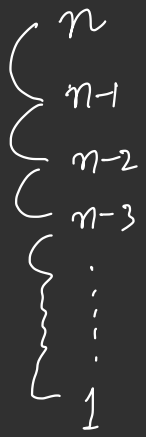


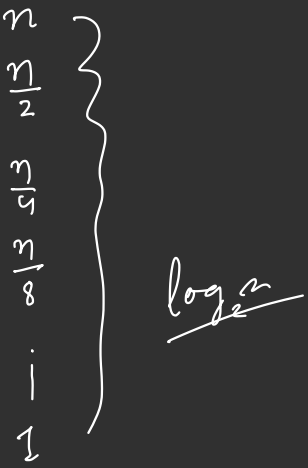
Let’s try to analyze a better approach

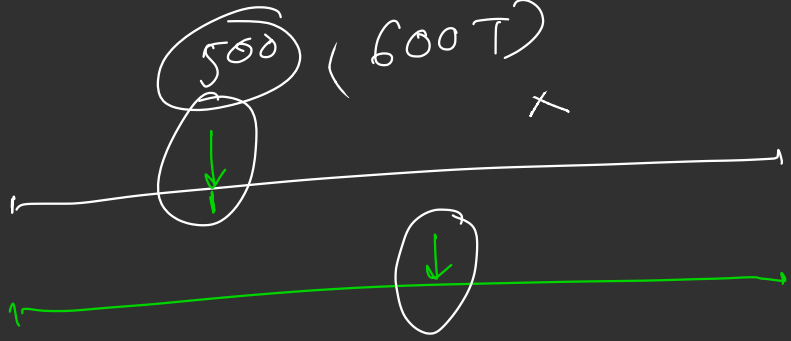


What are the number of operations required
in the worst case scenario?

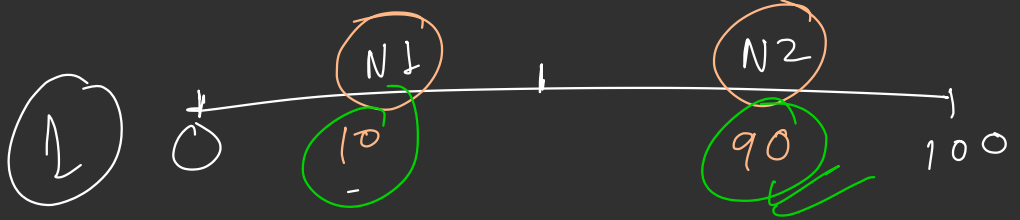
$1 \rightarrow 10^6$





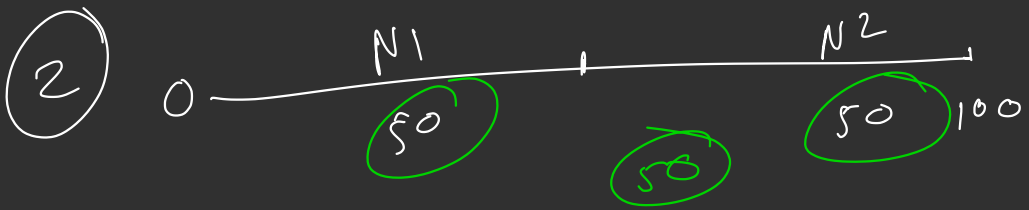


1	20	18	12	30	10	6	15	70
0	1	2	3	4	5	6	7	8

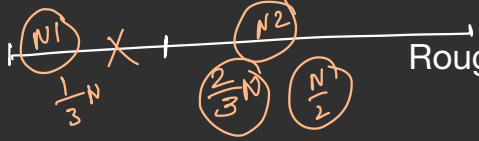


How to choose x?

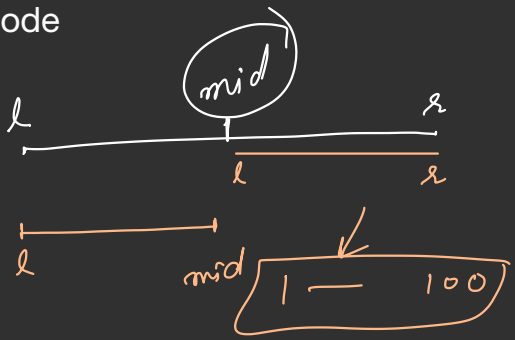
Target : Minimise max(N1, N2) where we know that N1 + N2 = N:



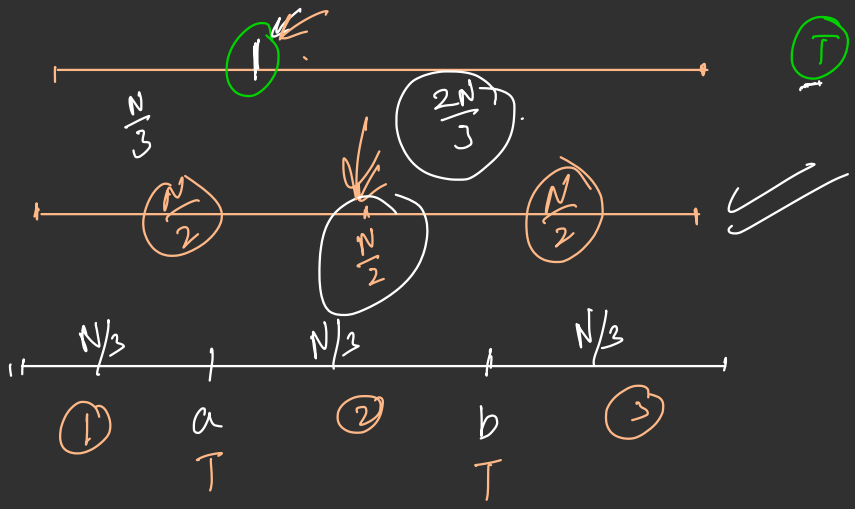
What we just discovered is
exactly what BINARY SEARCH is!

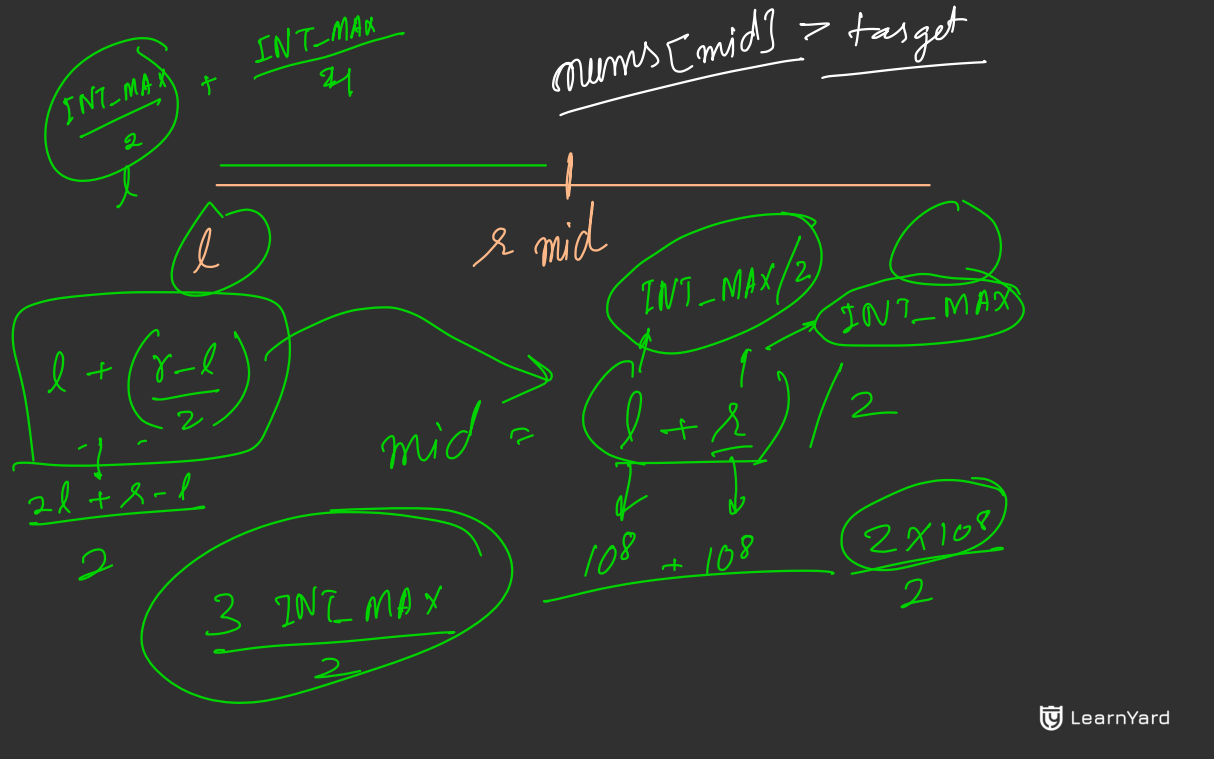


Rough Code




```
int findSecretNum(int l, int r) {
    if(l == r)
        return l;

    int mid = (l + r)/2;

    // Assume that isSNgreater() is a function that takes O(1) time
    bool checkForGreater = isSNgreater(mid);

    if(checkForGreater)
        return findSecretNum(mid + 1, r);

    return findSecretNum(l, mid);
}
```




INT_MAX/2 + INT_MAX/2
nums[mid] > target
l
r mid
l + (r-l)/2
mid = (l + r)/2
INT_MAX/2
INT_MAX
2l + r - 1 / 2
3 INT_MAX / 2
10^8 + 10^8
2 X 10^8 / 2

$-\infty - (-\infty/2)$

$l = \frac{-\infty}{2}$ $\quad r = -\infty$

$-\infty + \infty/2$

$= \frac{-\infty}{2}$

$\frac{3\infty}{2}$ $\quad \frac{l+r}{2}$

$\frac{\infty}{2} + \frac{\left(\frac{\infty}{2}\right)}{2}$

$l + \left(\frac{r-l}{2}\right)$ $\quad \frac{2l+r-l}{2}$

$= \frac{l+r}{2}$

$\frac{2}{4}\frac{\infty}{2} + \frac{\infty}{4}$

$\frac{3\infty}{4}$

Time Complexity?

Contacts Example

Dictionary Example

Assembly Queue Example

Git Bisect Example

Searching for a given target in a sorted Array

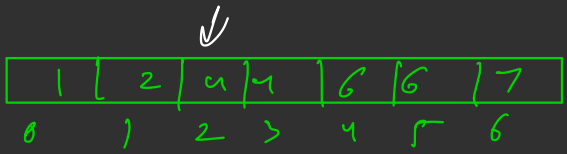
Power of Binary Search

Time Comp / Input Size	O(LogN)	O(√N)	O(N)	O(NLogN)	O(N*N)	O(N*N*N)	O(2^N)
20	~20 ns	~22 ns	~100 ns	~450 ns	~2 µs	~40 µs	~2 ms
50	~30 ns	~35 ns	~250 ns	~2 µs	~12 µs	~625 µs	~ 2 months
500	~45 ns	~111 ns	~3 µs	~25 µs	~1 ms	~650 ms	Out of Syllabus xD
5000 (5*10^3)	~60 ns	~350 ns	~25 µs	~300 µs	125 ms	~11 mins	Out of Syllabus xD
1 million (10^6)	~100 ns	~5 µs	~5 ms	~100 ms	~1.5 hours	~159 years	Out of Syllabus xD
100 mil (10^8)	~135 ns	~50 µs	~500 ms	~13 secs	~1.6 years	~159 megayears	Out of Syllabus xD
1 billion (10^9)	~150 ns	~0.2 ms	5 secs	~2.5 mins	~159 years	~159 eons	Out of Syllabus xD
1 trillion (10^12)	~200 ns	~5 ms	~1.5 hours	~5.5 hours	~159 megayears	~159 billion eons	Out of Syllabus xD
10^15	~250 ns	~200 ms	~2 months	~8 years	~159K eons	Out of Syllabus xD	Out of Syllabus xD
10^18	~300 ns	~ 5 secs	~159 years	~95 centuries	~159 billion eons	Out of Syllabus xD	Out of Syllabus xD

Approximate Time Taken
(assuming 2*10^8 operations per second)

Can we do better than LogN (base 2)?

Let’s implement binary search now



> Upper Bound & Lower Bound >=

3

target

4

Searching in a rotated sorted Array

Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!