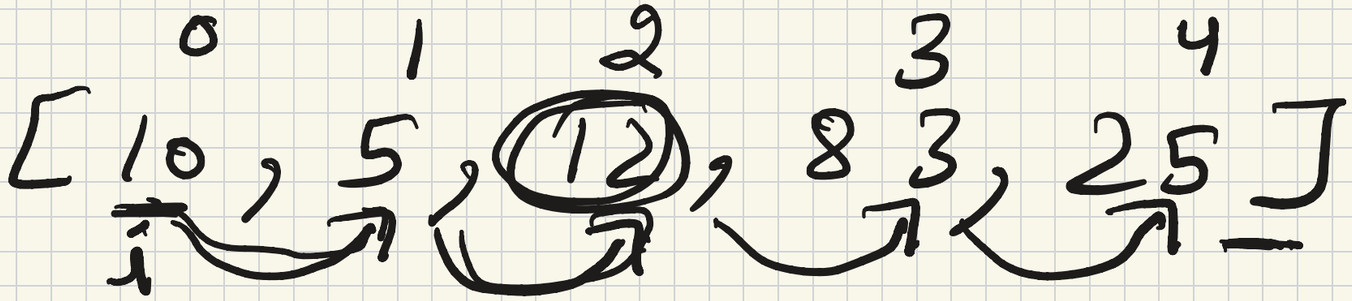
Linear Search ⇒

arr = [10, 5, 12, 83, 25]

Question ⇒ Weather 12 exist in array or not

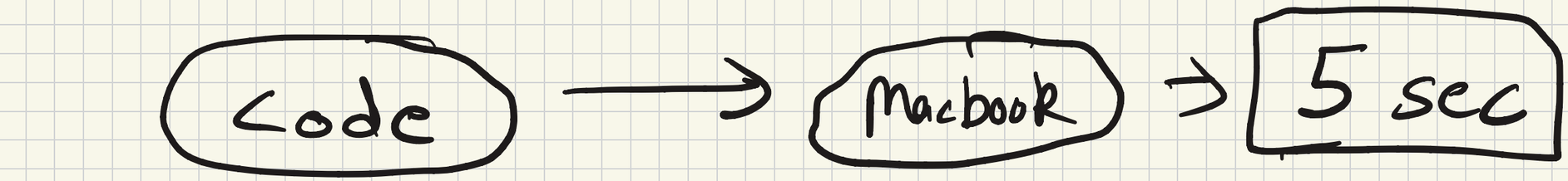


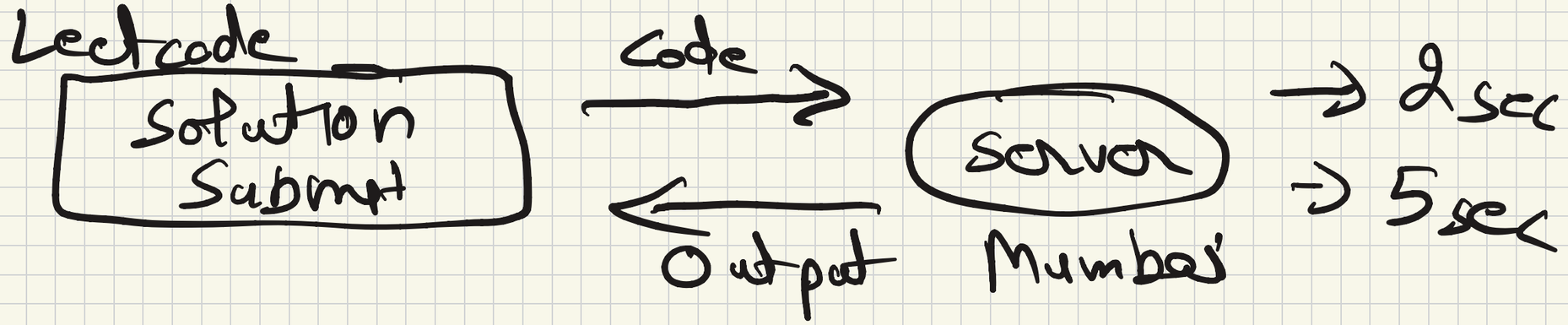
Target ⇒ 20

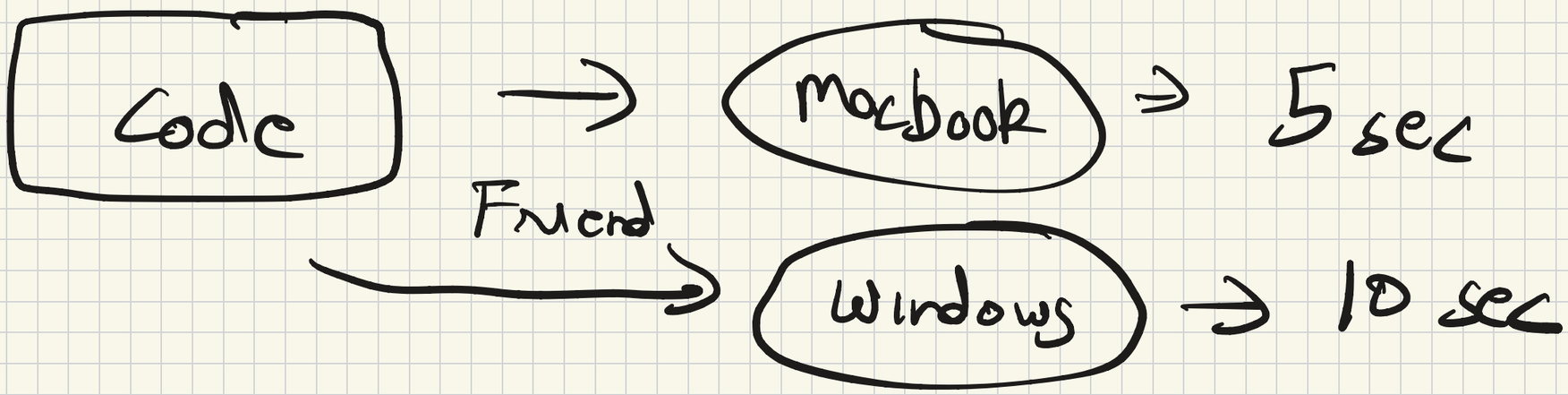
⇒ Return -1 if it does not exist.

Time & Space Complexity ⇒

Time Complexity → Time taken by my code to run

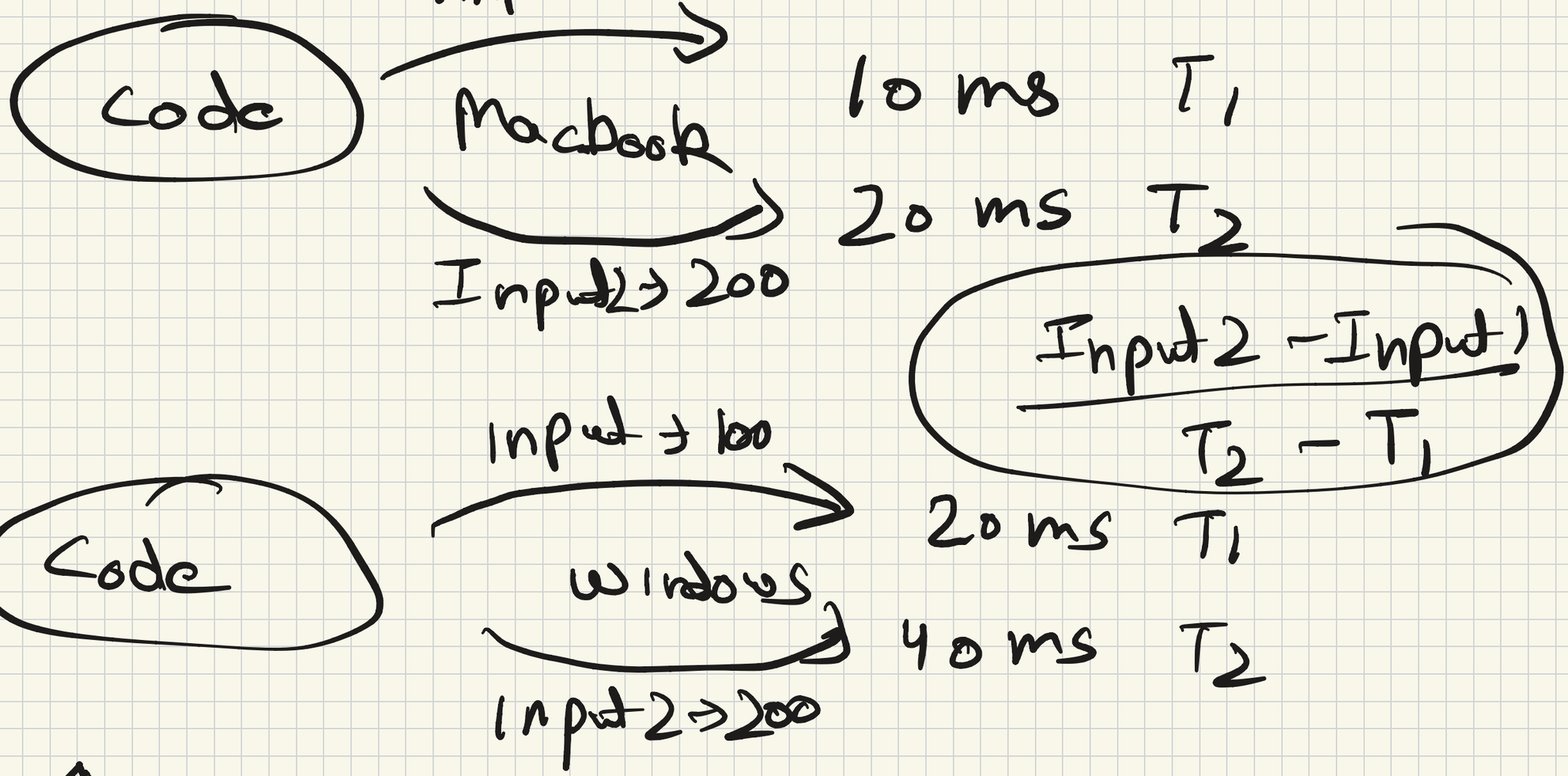






Time Complexity != time taken

Time Complexity ⇒ How does your time changes when your input grows



⇒ $\underline{O(N)}$

[10 , 12 , 8 , 26 , 53 , 42]

Best Case → 10 → 0

[10 , 12 , 8 , 26 ---------- 200 items]

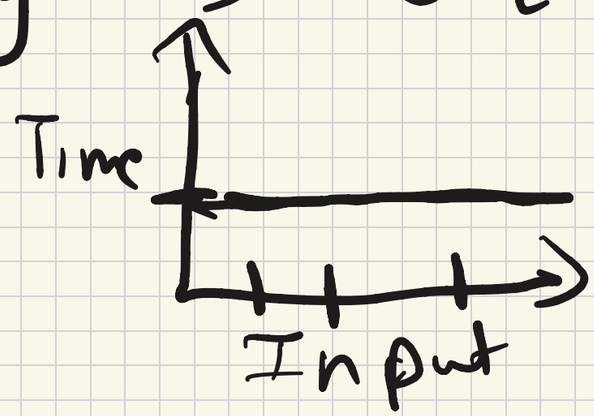
Target → 10 → 0

[10 , 12 , 8 , 26 ---------- 500 items]

Target → 10 → 0

Comparison → 1

Best complexity → O(1) Constant TC



1 ms

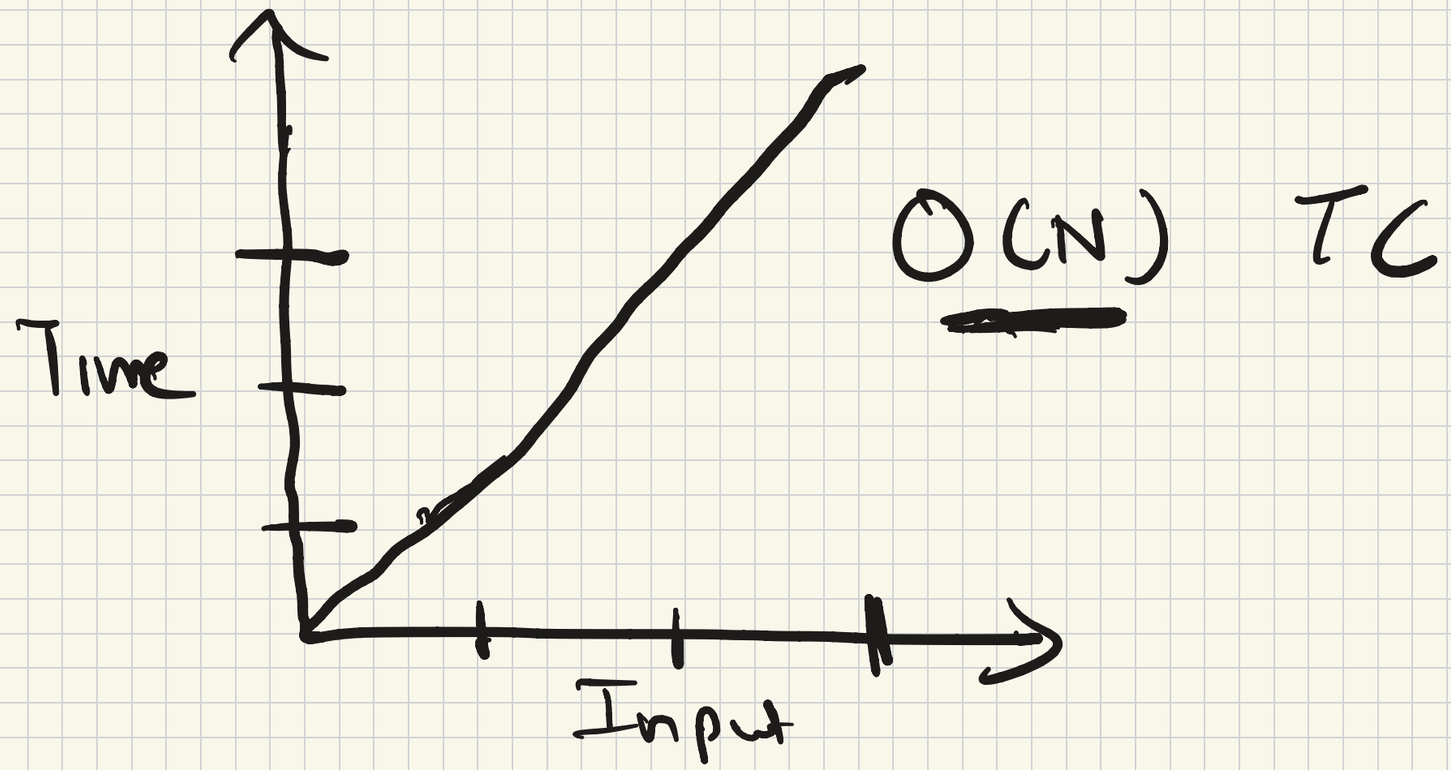
Worst case → [10, 12, 8, 15, 42, 83, 5]

Target → 21

comparison 7

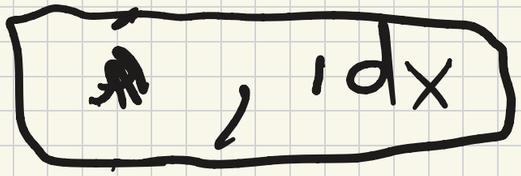
[------ 200 items] Comparison → 200

[------ 500 items] Comparison → 500



Linear Search → O(N) TC

Space Complexity → Extra space to solve my problem



Space

[100 items] → 2

[200 "] → 2

[500 "] → 2

O(1) SC

Constant

*

Search in a string → "@@@bham"

index

for → (i → str.length())

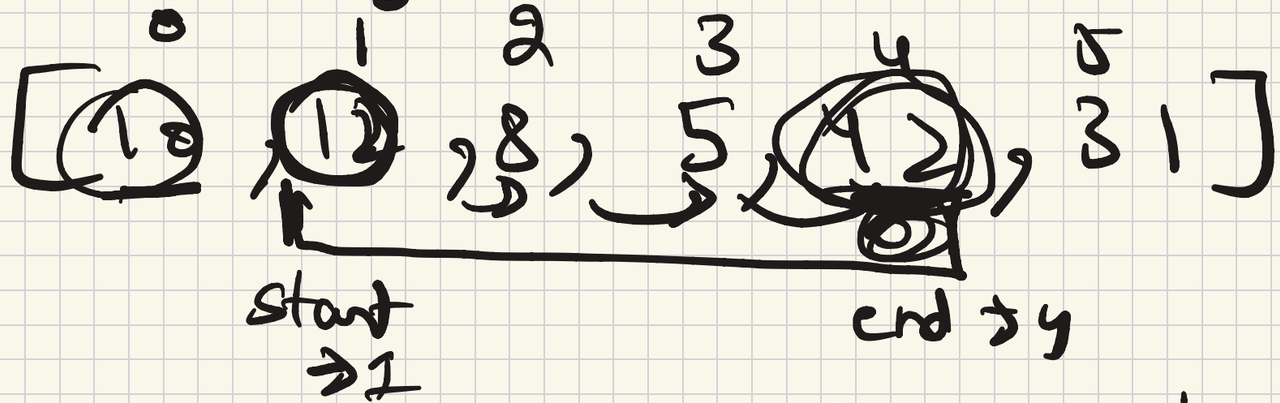
if (str.charAt(i) == target)

return i

3

return -1

✱ Search in a range :



Pseudo Code :-

return → -1

```
if (start > end) return -1
(end < arr.length)
for (i = start ; i <= end ; i++) {
        if (arr[i] == target) return i
    }
return -1
```

✱ Search in 2d array :

```
[ [10, 12, 8],
  [5, 15, 10, 8],
  [42, 31, 18, 16] ]
```

[[10, 12, 8] , [5, 15, 16, 11] , [42, 82]]

0 1 2 arr

for → (i → 0 → arr.length → i++)

for → (j → 0 → arr[i].length → j++){

if (arr[i][j] = Row

== target) 0 →

return [i, j] 1 →

2 →

3

return [-1, -1]

10 → 0, 0

15 → 1, 1

42 → 2, 0

11 → 1, 3