arr = [18, 12, 9, 14, 77, 50] size = 6

0 1 2 3 4 5

i i i i

→ Unsorted

Q: Find whether 14 exists in array or not.

If no value found, return -1.

Time Complexity:

Best: O(1) // Constant

Worst Case: O(N)

N ⇒ size of array

How many checks will the loop make in best case i.e. element found at 0^{th} index?

arr = [8, 9, 12, 18, ... 200 elements]
(0, 1)

target = 8

→ 1 comparison in best case.

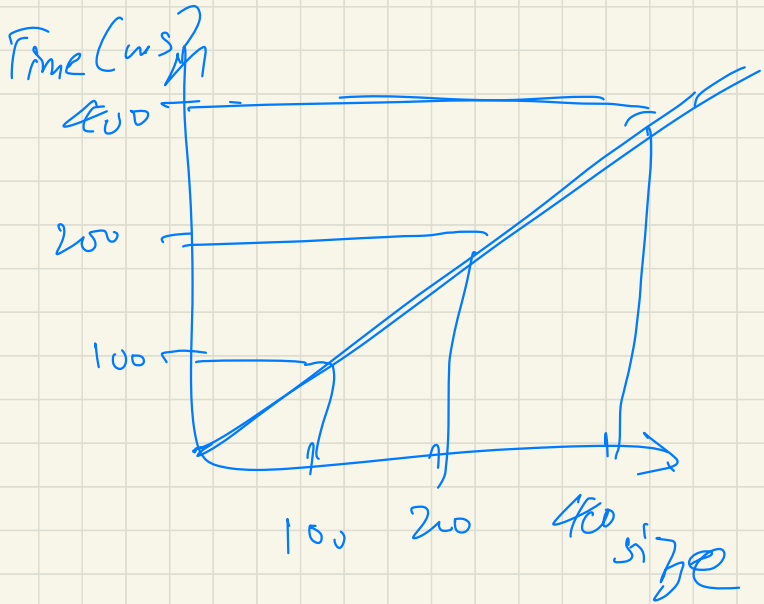
arr is now of size 1 lakh.

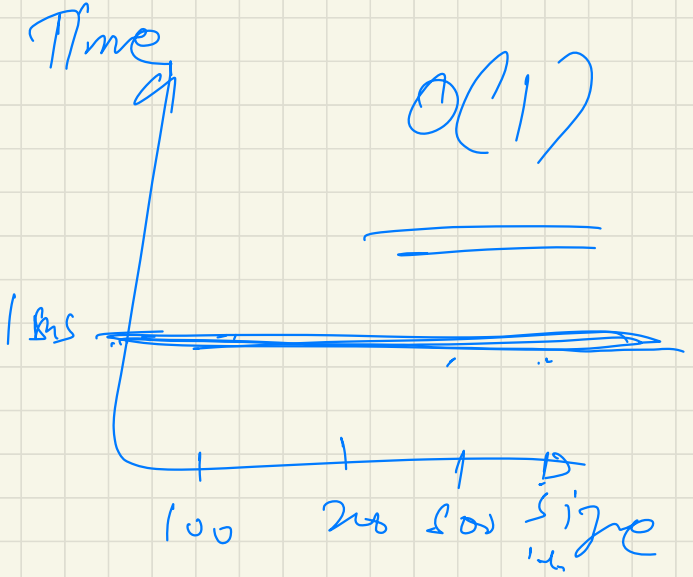
arr = [18, 12, 9, 7, ... 1 lakh items]
(0, 1)

target = 18. answer ⇒ only 1 comparison make. in best case.

Worst Case: You do not find the target item. iterate / go through every item and then in the end it says I did not find it.

size of array = 100 ⟹ 100 comparisons = 100ms
200 ⟹ 200 comparisons = 200ms
1 lakh ⟹ 1 lakhs = 100000 ms





Best case

worst case

500 items ≅ 500 ms

arr = [18, 12, -7, 3, 14, 28]

(indices: 0, 1, 2, 3, 4, 5)

Q: Search for 3 in the range of index [1, 4]

Q. Find min element in the array.

arr = [18, 12, -7, 3, 14, 28]

Min

-7

arr = [[1, 2, 3], [9, 18, 5], [6, 7, 14]]

```
for( row=0; row<len(arr)
    row++) {
    for(c=0; c<len(r),
        c++) {
        if(arr[r][c] ==
            target)
            //found ans
    }
}
```

max

~~-2147483648~~

~~1~~ ~~2~~ ~~3~~ ~~9~~ 18

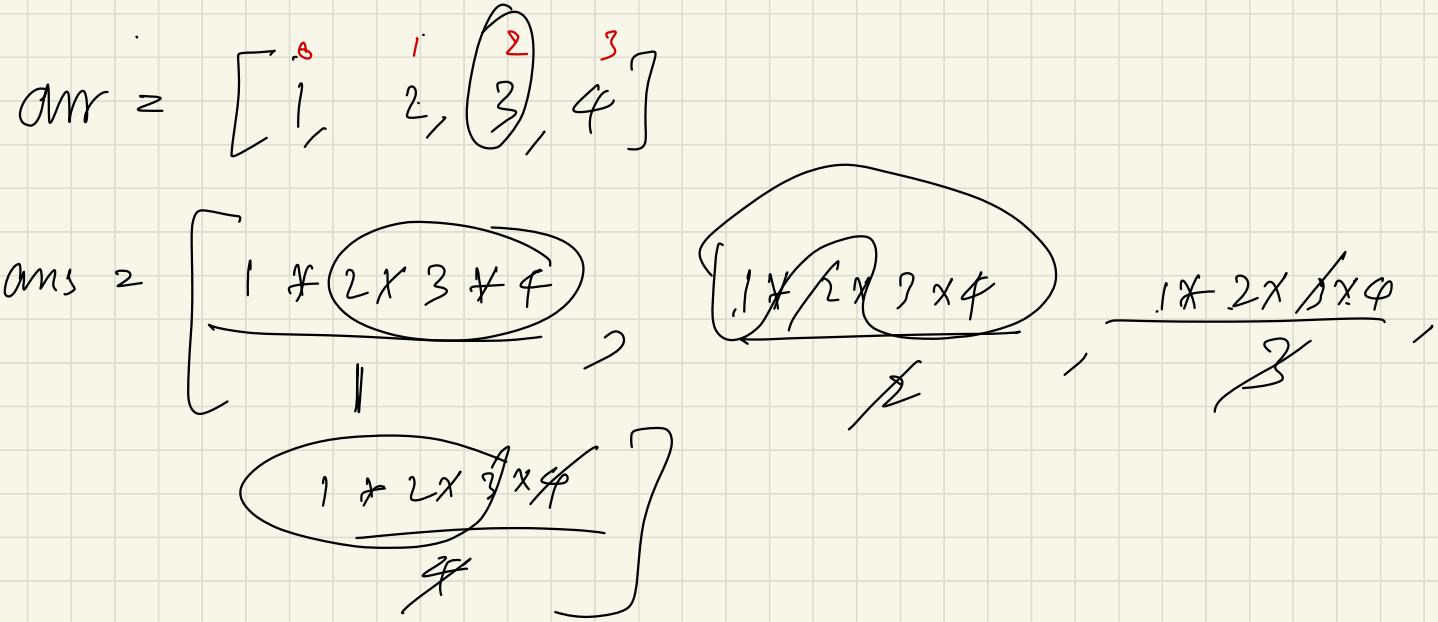
$arr = [1, 2, 3, 4]$ (0, 1, 2, 3)

$ans = [2*3*4, 1*3*4, 1*2*4, 1*2*3]$

$total = 1*2*3*4 = 24$

$ans = \left[\frac{24}{1}, \frac{24}{2}, \frac{24}{3}, \frac{24}{4}\right]$

$= [24, 12, 8, 6]$


arr = [1, 2, 3, 4]
0 1 2 3
ans = [1 * 2 * 3 * 4 / 1, 1 * 2 * 3 * 4 / 2, 1 * 2 * 3 * 4 / 3, 1 * 2 * 3 * 4 / 4]

Q Find no. of nos. that have even no. of digits.

nums = [18, 124, 9, 1764, 98, 1]

Ans = 3

(1) Count the no. of digits

(2) Convert 1764 => "1764"

take the length

count = 0 1 2 3 4

1764

176

17

1

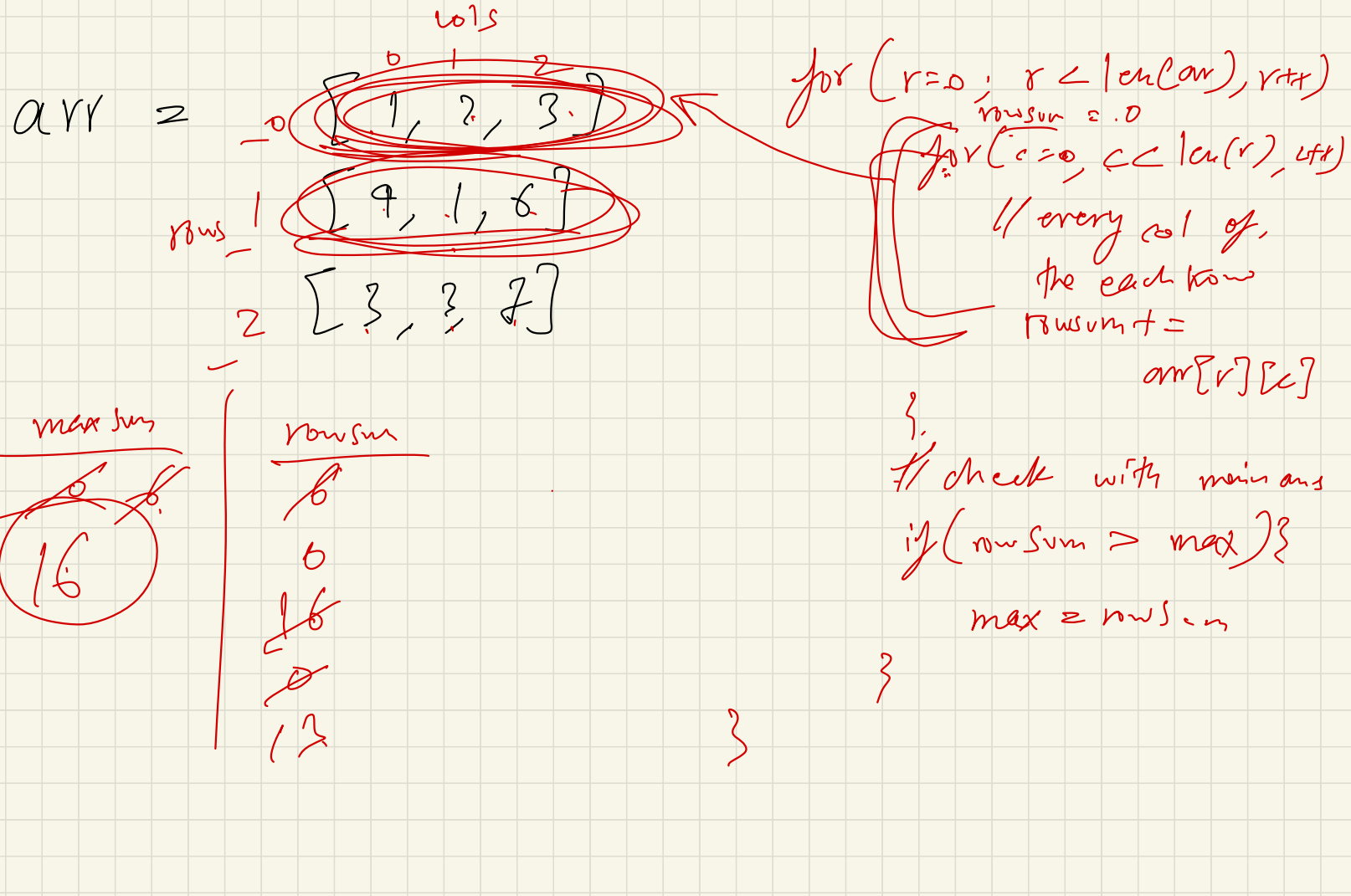
0

while (n > 0) {

count++

n = n/10

}


cols
0 1 2
arr =
0 [1, 2, 3]
rows 1 [9, 1, 6]
2 [3, 3, 7]
for (r=0; r < len(arr), r++)
rowsum = 0
for (c=0, c < len(r), c++)
// every col of the each row
rowsum += arr[r][c]
}
// check with max ans
if (rowSum > max){
max = rowSum
}
}
max sum
16
rowsum
6
0
16
0
13