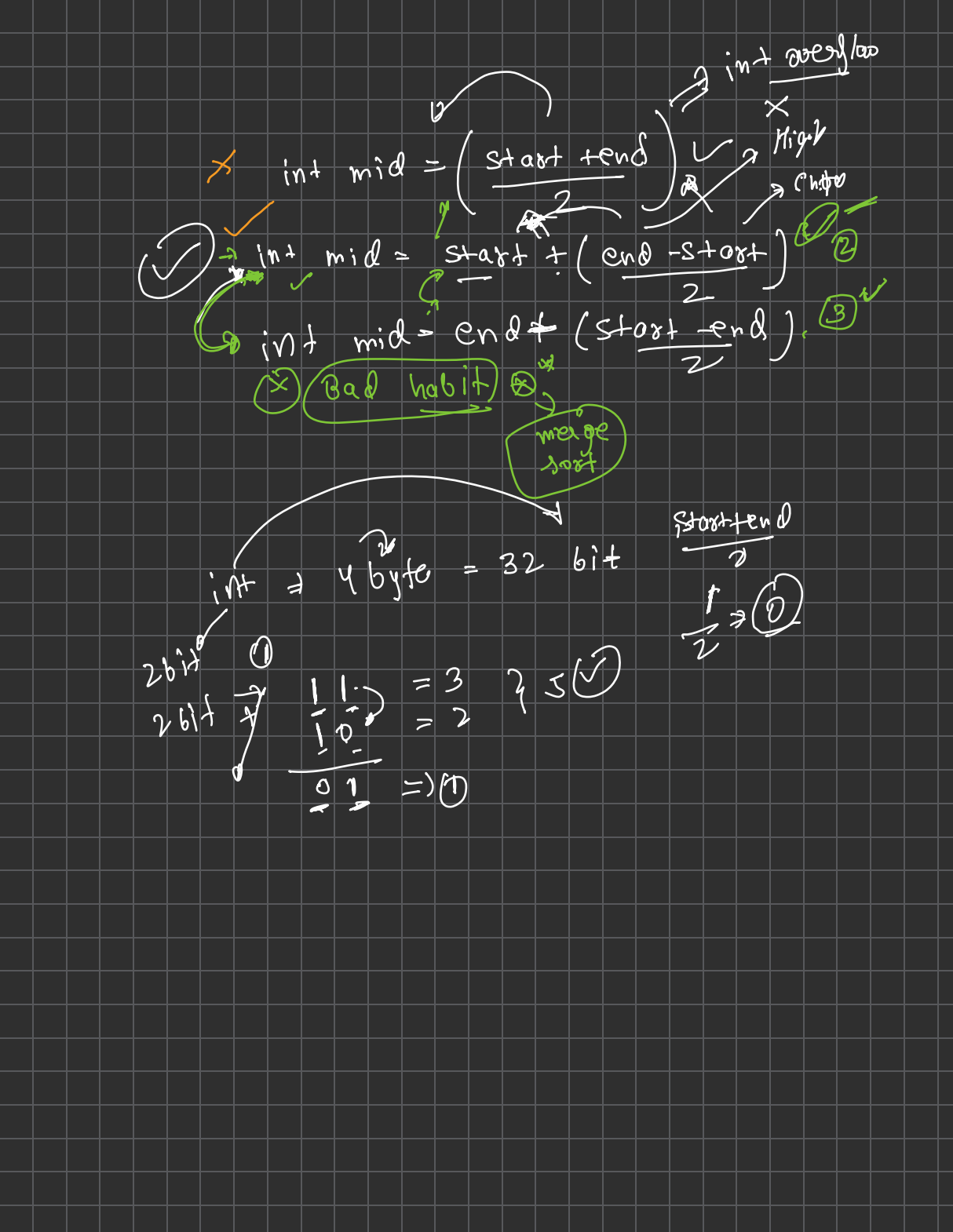
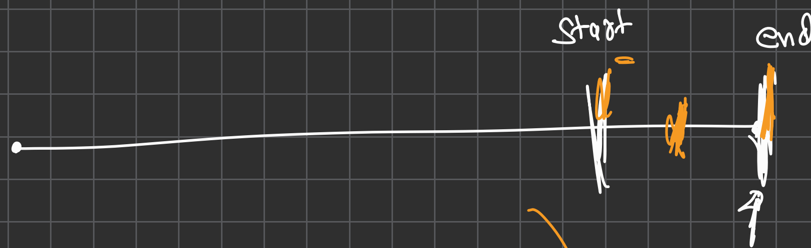

int mid = (start + end) / 2
int overflow
Right
Right
int mid = start + (end - start) / 2
int mid = end + (start - end) / 2
Bad habit
merge sort
int ⇒ 4 byte = 32 bit
start+end
2
2 bit
2 bit
1 1 = 3
1 0 = 2
0 1 ⇒ 1
3 ⇒ 2
2 ⇒ 1

$$\frac{2^{31} + 2^{31} - 1}{2} \Rightarrow \text{integer}$$



$$mid \Rightarrow \left(start + \frac{(end - start)}{2} \right)$$

int ⇒ 4 bytes ⇒ 32 bit (×8)

start + end → fit ✗ → int overflow

int = 2 bit (1) ← store

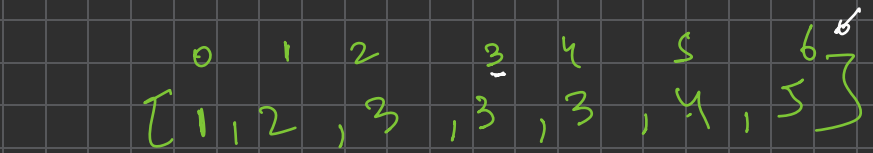
11 → 3 ✗

10 → 2

01 → 1

example

int mid



* target = 3

start = 0

end = 6

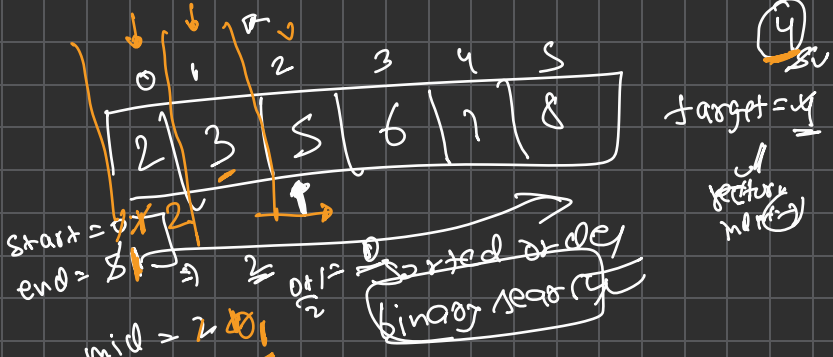
mid =

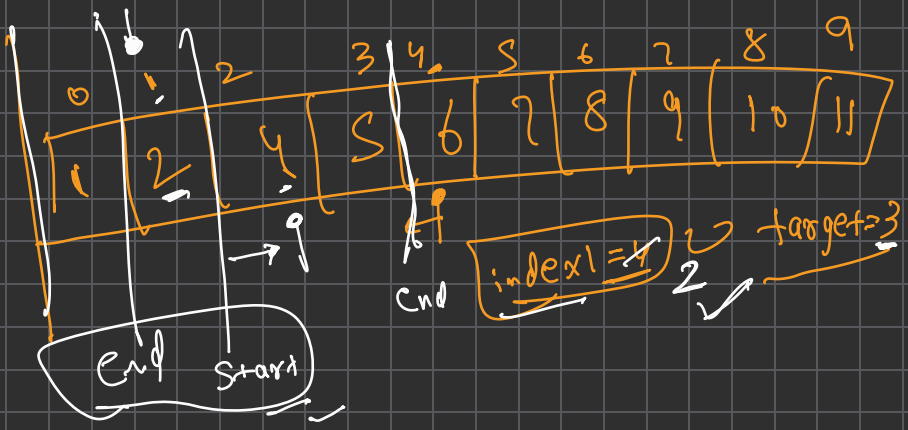
first index

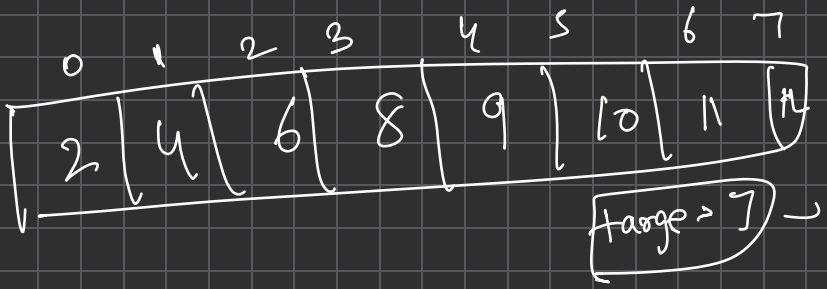
first index = 2

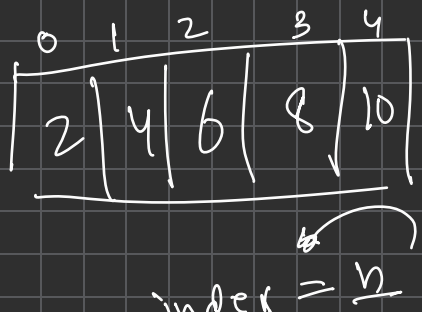
last index = 4

2 time Binary search




0 1 2 3 4 5 6 7 8 9
1 2 4 5 6 7 8 9 10 11
end
start
end
index1 = 4
target = 3


0 1 2 3 4 5 6 7
2 4 6 8 9 10 11 14
target = 7


0 1 2 3 4
2 4 6 8 10
s
arr.size
target=12
index = 2