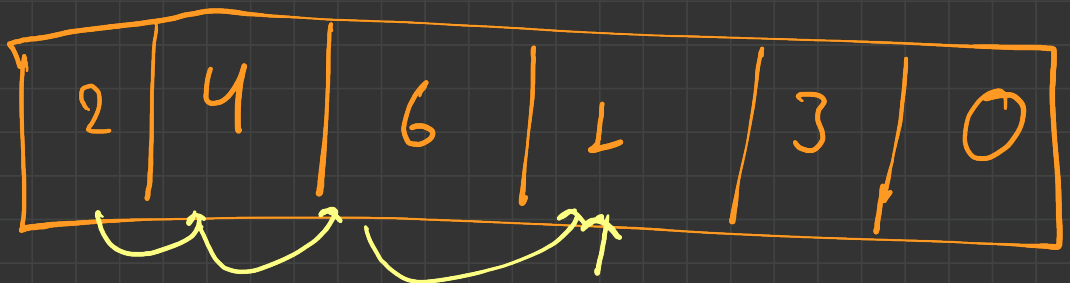
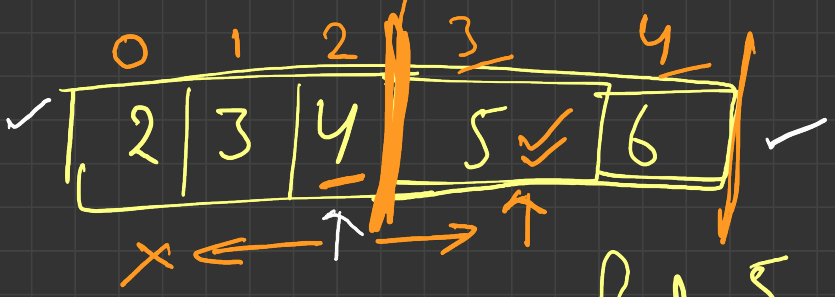
Day - 23

Binary Search



O(N)

Example



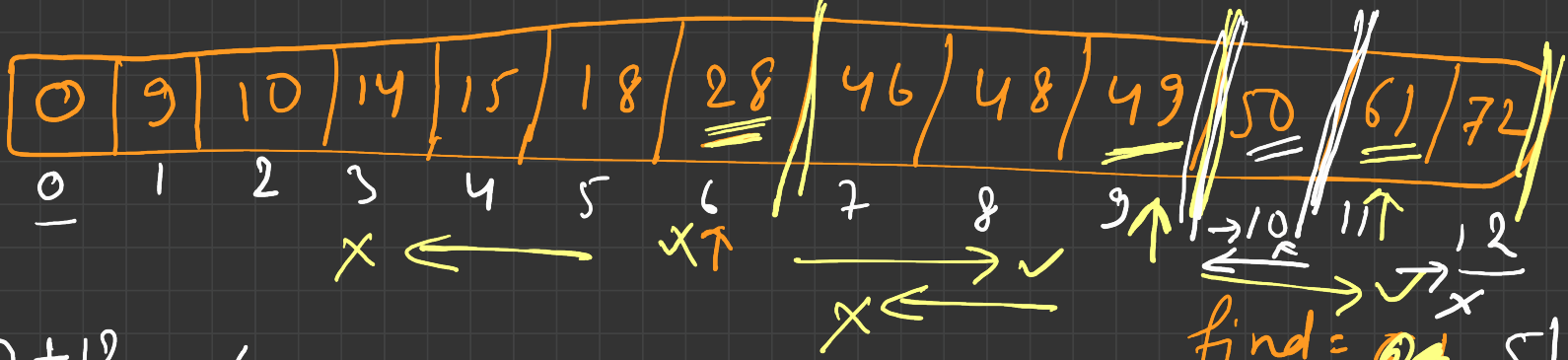
$\frac{0+4}{2} = \frac{4}{2} = 2$

2

find: 5

$\frac{3+4}{2} = \frac{7}{2} = 3$

$$M.I = \frac{1st\ index + last\ index}{2}$$

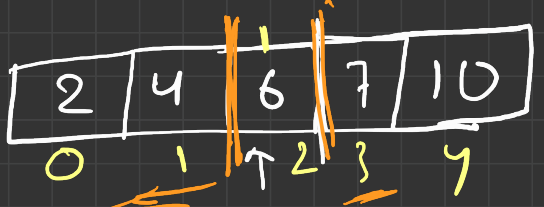


$\frac{0+12}{2} = 6$

$\frac{7+12}{2} = \frac{19}{2} = 9.5 = 9$

$\frac{10+12}{2} = \frac{22}{2} = 11$

$\frac{10+10}{2} = 10$



find = 7

$\frac{0+4}{2} = 2$

Start = 0

end = 4

① mid = $\frac{start + end}{2}$

6 > 5

6 < 7

② if (arr[mid] == Elem end)

cout << "find !!"

Break;

else if (arr[mid] < Elem end)

move to right side Start = mid + 1

else move to left side

end = mid - 1

```
start = 0
end = n-1
while (start <= end) {
    mid = (start + end) / 2
    if(arr[mid] == Element) {
        cout << "Finded";
        break;
    }
    else if (arr[mid] < Element) {
        start = mid + 1
    }
    else
        end = mid - 1
```

$\frac{N}{2^0} \rightarrow N = 1$

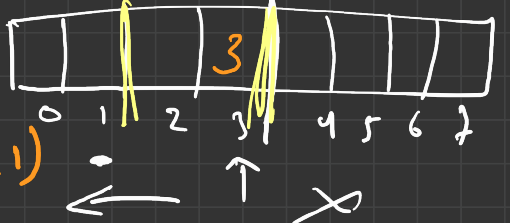
$\frac{N}{2^1} \rightarrow \frac{N}{2} = 1$ step

$\frac{N}{2^2} \rightarrow \frac{N}{4} = 1$ step $2+1=3$

$\frac{N}{2^x} \rightarrow \frac{N}{2^x} = 1$ step $x+1$ steps

Best $\rightarrow O(1)$

$\Rightarrow \Omega(1)$



Avg $= O(\log N)$

$\Theta(\log N)$

x

$\frac{N}{2^x} = 1$

$N = 2^x$

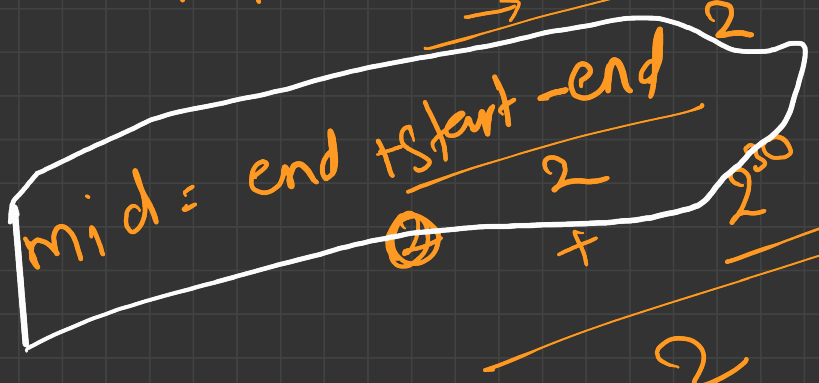
$\log N = x \log 2$

$x = \frac{\log N}{\log 2} = \log_2 N$

$O(\log_2 N)$

mid = start + end / 2

int → $2^{31} - 1$



$2^{20} + 2^{30}$ / 2

mid = start + (end - start) / 2

$2^{20} - 2^{20}$

start = 0

end = 8

$\frac{0+8}{2} = 4$

$\frac{1+7}{2} = 4$

$\frac{0+8-0}{2}$

$= 4$

$\frac{1+7-1}{2} = \frac{1+3}{2} = 4$

32 23 12 11 10 9 8 7 1 0 -1