→ Linear Search
→ Binary Search
* → interpolation Search
→ (ternary Search) → Adv

traversing →

{1, 5, 8, 10, 11, 12}

key → indx (8)

{1, 5, 10, 11, 12, 9}

traverse →

Linear Search

→ Acc O(1)
→ Update O(1)
→ get O(1)

→ arr = {1, 5, 8, 10, 11, 12}

(key = 8)

```
for( i:0 → n)
    if (key == arr[i])
        return i
```

linear Search

O(n)

O(1)

linear Search

Binary Search ([Search space] should be sorted) [{(Square Root) first occurence}]

↳ iterative –

↳ Recursive –

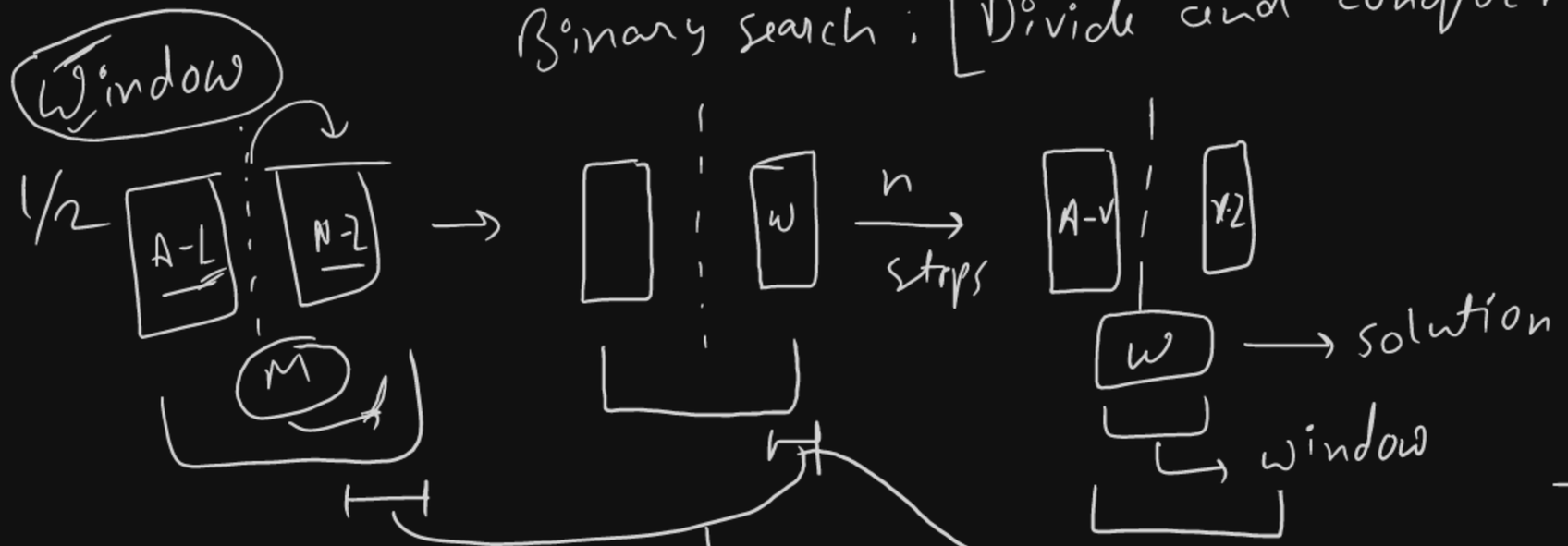
① Complexity / concept

[→ dictionary :

↳ Search Space

Binary search : [Divide and conquer]

make problem smaller untill you reach to the soln

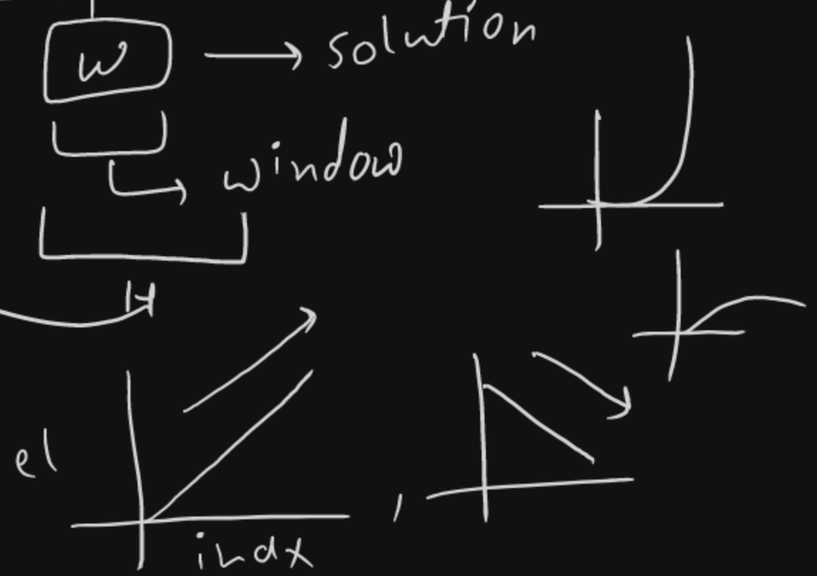


[Sorted]

[→ monotonic function (same rate)

arr = {1, 3, 5, 8} – sorted

= {8, 5, 3, 1} –

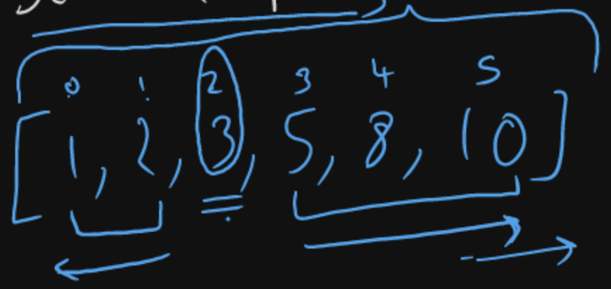


Complexity – $O(\log n)$

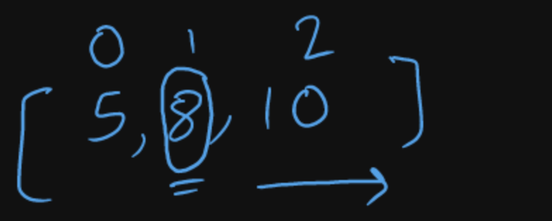
$O(1) < [O(\log n)] < [O(n)]$

[search space] – array → [1, 2, 3, 5, 8, 10] (12)

$\overset{0}{1}, \overset{1}{2}, \overset{2}{3}, \overset{3}{5}, \overset{4}{8}, \overset{5}{10}$



$\xrightarrow{12}$ BS() ✓



$5/2 = 2$

$\overset{0}{5}, \overset{1}{8}, \overset{2}{10}$

$\xrightarrow{12}$ BS() ✓

$3/2 = 1$

$\overset{0}{10}$

$\xrightarrow{12}$ BS() ✓

$1/2 = 0$

[12 is not present]

5

itr(BS) itr(LS)

3 5 BS() → (1/2) ✓ → (search)

$O(n)$

$\to O(n/2)$ +

$\to O(n/4)$ +

$\to O(n/8)$ +

$\to O(n/16)$ +

$\vdots$

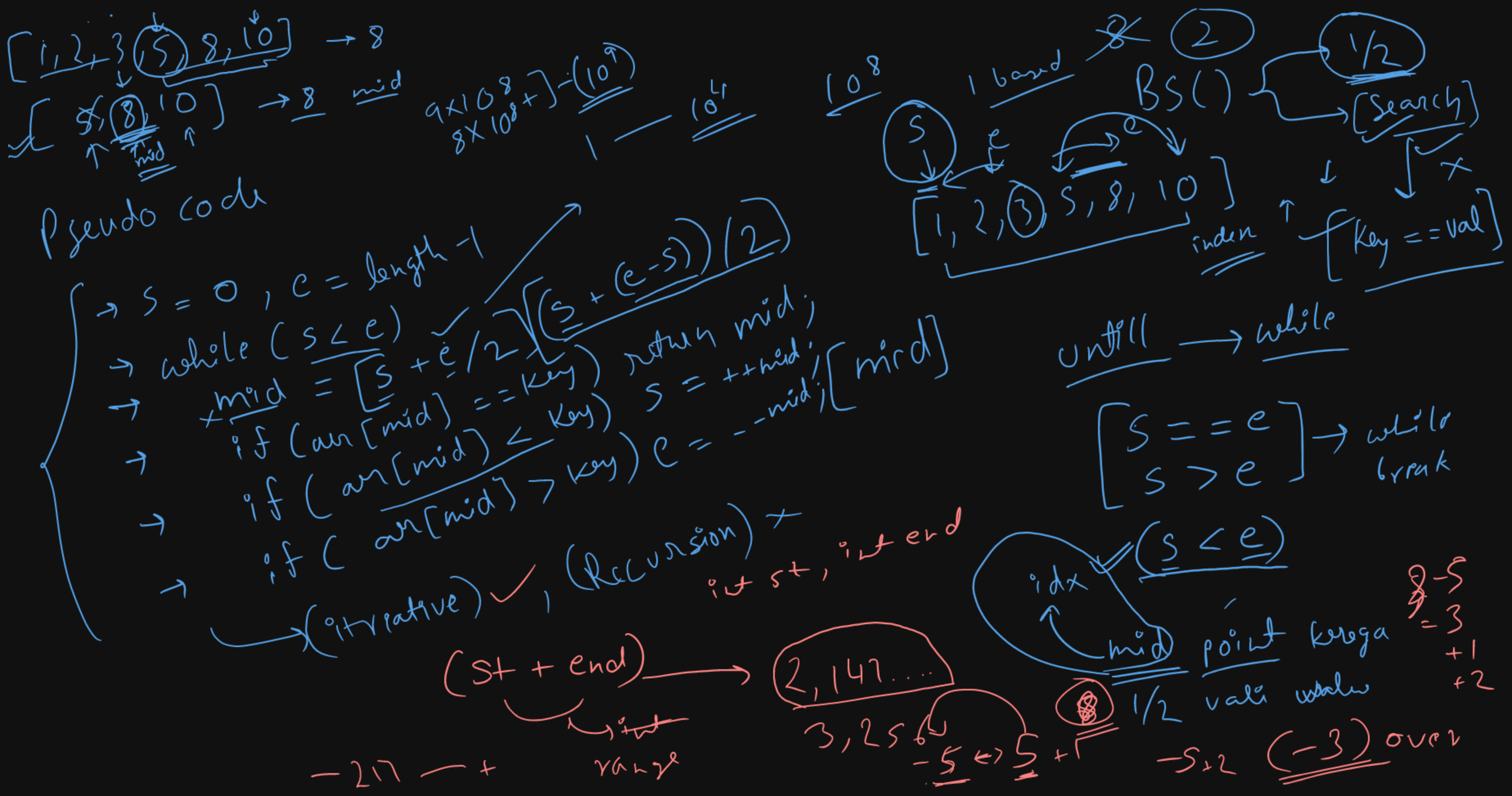
$\to O(n/2^k)$

n

$n\left(\frac{1}{2}+\frac{1}{4}+\frac{1}{8} \cdots \frac{1}{2^k}\right)$

$(\log_2 n)$

(Akra. Bazzi)



[1, 2, 3, 5, 8, 10] → 8

[5, 8, 10] → 8 mid

mid

9×10^8
$8 \times 10^8 +$] - (10^9)

1 — 10^4

10^8

1 based ~~8~~ 2

BS() → 1/2 → [Search]

s e e

[1, 2, 3, 5, 8, 10]

index ↑ [Key == val]

Pseudo code

→ s = 0, e = length - 1

→ while (s < e)

→ mid = [s + e/2] [(s + (e - s))/2]

→ if (arr[mid] == key) return mid;

→ if (arr[mid] < key) s = ++mid; [mid]

→ if (arr[mid] > key) e = --mid;

→ (itreative) ✓, (Recursion) ×

int st, int end

(st + end) → 2,147....

~~int~~ range

3,256

-217 — +

-5 ↔ 5 +1

-5 +2 (-3) over

untill → while

[s == e
s > e] → while break

(s < e)

idx

mid point kerega

8 1/2 vali value

8 - 5
= 3
+1
+2