

Binary Search why??

Sorted array

Increasing / Decreasing

Sorted array

0	1	2	3	4	5
2	4	6	12	14	16

$x = 14$

output = 4

low = 0

high = 5

while (low <= high) {

mid = (0 + 5) / 2 = 2 (low + high) / 2;

if (arr[mid] == x) {

return mid;

else if (arr[mid] < x) {

low = mid + 1;

Right side search

else {

high = mid - 1;

Left side search

x = 4

$$\underline{\text{mid}} = (\text{low} + \text{high}) / 2 ; \quad \text{--- (1)}$$

Recommended

$$\underline{\text{mid}} = \text{low} + (\text{high} - \text{low}) / 2 \quad \text{--- (2)}$$

Analyser:-

$$\frac{N}{2^0} + \frac{N}{2^1} + \frac{N}{2^2} + \dots + 1$$

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

$$N = 2^k$$

$$\log_2 N = k \log_2 2$$

$$k = \log_2 N$$

# iterations

Time complexity

$$\hookrightarrow \underline{O(\log_2 N)}$$

Space complexity

$$\hookrightarrow \underline{O(1)}$$

Linear Search  
↳  $O(n)$

Binary Search  
↳  $O(\log n)$

Advantage

↑  
Data is very huge

Disadvantage

↳ Sorted data