Binary Search

↳ sorted array
↳ ascending / Descending order

sorted

	0	1	2	3	4	5	6
arr →	2	4	6	8	12	14	20

target = 4

low = 0
high = 2
mid = $\frac{0+2}{2}$ = 1

low = 0
high = 6

target = 14
low = 4
mid = $\frac{4+6}{2}$ = 5

mid = $\frac{(0+6)}{2}$

$\boxed{mid = 3}$

```
while( low <= high) {
        if (arr(mid) == target)
              return mid;
        else if( arr(mid) < target)
                 low = mid+1;
        else {
              high = mid -1;
        }
}
```

Time complexity

search space

$\hookrightarrow \frac{N}{2^0} + N/2^1 + N/2^2 + \dots + 1$ $\Big\}$ k times

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

$$N = 2^k$$

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

$$\boxed{K = \log_2 N}$$

$\hookrightarrow$ Time complexity of a binary search

overflow

$$\boxed{mid = \frac{low + high}{2}}$$

$$\boxed{mid = \frac{low + (high - low)}{2}}$$

$$\boxed{\text{Space complexity} - O(1)}$$

Advantage

↳ Huge datasaet

↳ Easily manegeeble

Logically sorted

Not sorted

Disadvantage

↳ Sorted fashion

mid = 5

low = 4

high = 7

0	1	2	3	4	5	6	7
2	1	7	10	12	∞	∞	∞

Position of first infinite

Output = 5

modified binary search

target = ∞

$$mid = \frac{0+7}{2} = 3$$