-: Binary Search :-

→ Sorting : To follow order in a fashion.
- Increasing Order → {2, 2, 3, 4, 4, 5}
- Decreasing Order. → {11, 11, 9, 9, 4, 2, 1}

→ These types of array called as sorted array.

→ Binary Search
- This is a two way searching algorithm.

Q → {(2, 4, 11), 20, 37} → Sorted Array.
0 1 2 3 4

(11 != 20)

- Then it will search in increasing order in increasing sorted array.

mid index = (Start + End) / 2

(0 + 4) / 2 = 2.

```
if (arr[mid] == target)
    {
        return mid;
    }
else if (arr[mid] < target)
    {
        start = mid + 1;
    }
else
    {
      end = mid - 1;
    }.
```

→ B.S. :-

- Binary search will work only on Sorted Data Structure.
- Binary search is very fast to find an element

$$\Rightarrow \frac{N}{2^K} = 1$$

$$N = 2^K$$

$$K = \log_2 N$$

$N \to \frac{N}{2} \to \frac{N}{4} \to \frac{N}{16}$

$\frac{N}{2^K} = 1.$

- It works on Ranges as well.

→ To overcome the overflow Problem (Ranges of variable)

$$mid = \frac{Start + End}{2} \times$$

$$mid = Start + \frac{End - Start}{2}$$ [Right approach]

→ Array Sorted in Decreasing Order ↓

{ 87, 54, 36, 22, 18, 1 };
0 1 2 3 4 5

- Find Mid = Start + $\frac{\text{End} - \text{Start}}{2}$

= 2.

if (arr[mid] < target) → [Algorithm]
{
~~start~~ = ~~mid~~ +
end = mid - 1;
}
else ~~if~~ ()
{
start = mid + 1;
}

→ Order-Agnostic Binary S ←

→
- Sorted array - - - - - - - - - - - - - - -
- Search
- Array
- Target.

Q→ How do find that is an array in increasing order or Decreasing Order.

- Order-Agnostic Binary Search.
  - S = E [All elements are Same]
  - S > E [Decreasing Order]
  - S < E [Increasing Order]