BINARY SEARCH

1) What is Binary search ? ⇒ Elemination ½ part from the searching size

2) ~~What is~~ Why Binary search ⇒ What's the problem ~~for~~ in Linear search.

Linear search ⇒ $O(n)$
Binary search ⇒ $O(\log(n))$] → for sorted array.

Linear search ⇒ $O(n)$
Binary search ⇒ $O(n\log(n) + \log(n))$] - for unsorted array

~~Can we~~ Can we apply B.S on unsorted array, ~~Why~~ How
⇒ Yes, By sorting It, then apping B.S.
Why ? (Time complexity of Binary search > Linear search)
⇒ If we have 'q' queries then we use BS

Linear search ⇒ $O(n \times q)$
Binary search ⇒ $O(n\log(n) + q\,(\log(n))$] - for unsorted array.

3) When to Use Binary search (intution for Binary search) ⇒ It will not provided in question to use B.S

⇒ If function is monotonic

⇒ If (maximum / minimum) value is to be find

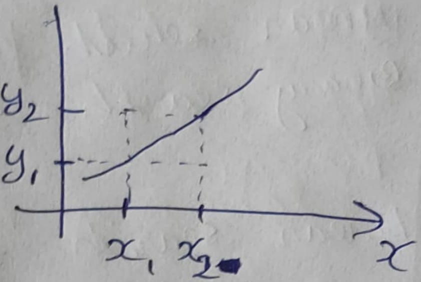
4) Ok, what is monotonicity ⇒

If function is increasing / decreasing [not necessary strictly ↑/↓]

How we can find monotonicity

i) $y_1 = f(x_1)$

$y_2 = f(x_2)$



If $(y_2 - y_1)$ is always greater/smaller than 0. then monotonic

ii) Derivation.

$y = f(x)$

$$\frac{dy}{dx} = \frac{d(f(x))}{dx}$$

If the derivative is always increasing / decreasing function for particular domain

↳ input constraints.

APPLYING BINARY SEARCH

⇒
- Distint sorted array.
- Find if element is present or not.
- Variation ⇒ Position of element. on the array.

```
res = -1
low = 0 , High = n-1 ;          // initialization
while (low ≤ high)
{
    long long mid = low + (high - low) /2 ;
      If (a[mid] == key)   // a[mid] is compared to key
       {  res = mid;
           Break;
       }
     Else If (a[mid] < key)
       {  low = mid + 1 ;
       }
     Else
       {
          high = mid - 1 ;
       }
}
return res;     // if res = -1 ⇒ no element found
```

Errors in different lines of code.

X **Check function** ka use upper bound ke baabd batayenge

- lower bound ⇒ [illegible] Defination ⇒ find lowest index which is greater than or equal to key.

Implement using slight change.

⇒ **Use check function.**

. **See last line**

- Similarly of upper_bound.
- finding ferequency of elements.
- 1300 rated problems mai lower_bound. or upper bound ka use karke, solution accepted ho jayega

Now lower_bound ka use karke question (1) kar sakte (distint sorted array)