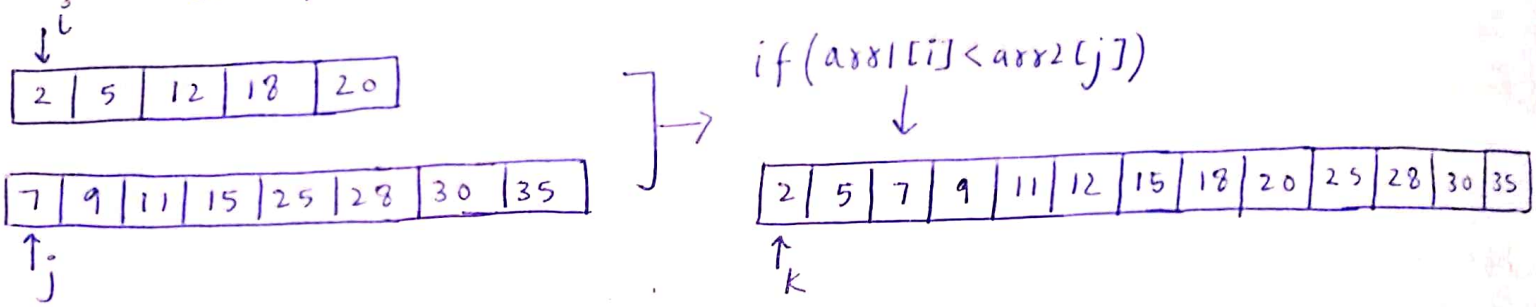
⑧ MERGE TWO SORTED ARRAY

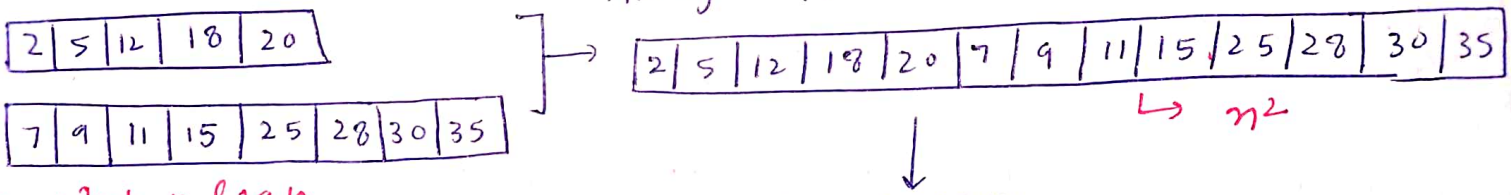
Hume ② Array (SORTED) given hai! (Integer Sorted Array)



∴ 2 < 7

∴ final [0] = 2 → i++, j++!

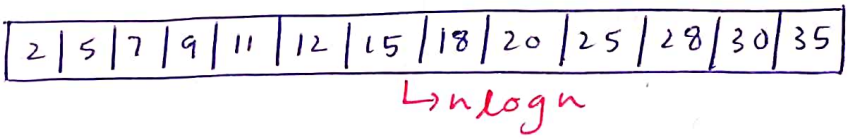
BRUTE-FORCE APPROACH



∴ $n^2 + n\log n$

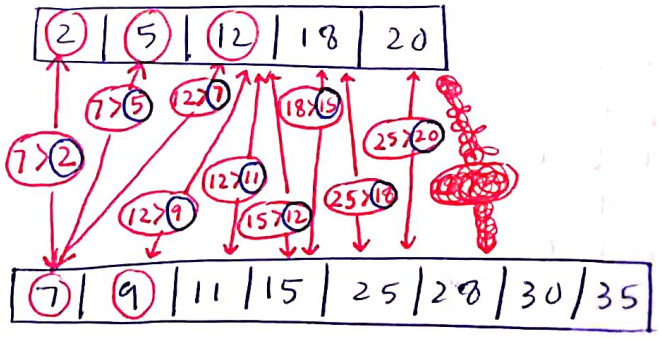
Overall $T(n) = n^2$

SORTED ARRAY



EFFICIENT APPROACH

* Hum ② new Array liven Hongay! (SORTED ARRAYES)
* Hum ek Naya Array banayengy equal to sum of ② Arrays.
* (i) pointer on Array 1
  (j) pointer on Array 2
  (k) pointer on Final Array
* Array ① aur Array ke bich me ek condition check lagega!
* Agar [Arr[i] < Arr[j]] then {Arr[k] = Arr[i]}
  if [Arr[i] > Arr[j]] then {Arr[k] = Arr[j]}
* Jab Array ① khatam Hojaye toh Array ② ke elements same to same merge krdo jitne bache hai!



∴ Final Array: 2 5 7 9 11 12 15 18 20 25 28 30 35

// Hum Dheere-Dheere saare array ko travel kr pa rahey hai Aur ek new Merged-Sorted Array bana rahay hai!
Jo ki sorted-order me hai (Array ① and Array ② se smaller value select krke new Array me daali)

```java
public static void main (String [] args) {
  Scanner s = new Scanner (System.in);
  int n = s.nextInt();
  int [] a = new int [n];
  for(int i=0; i<n; i++)
  {   a[i] = s.nextInt();
  }
  int m = s.nextInt();
  int [] b = new int [m];
  for(int i=0; i<m; i++)
  {
     b[i] = s.nextInt();
  }
  int [] mergedArray = mergeTwoSortedArray (a,b);
  print (mergedArray);
  }
public static void print (int [] arr)
{    for(int i=0; i<arr.length; i++){
         System.out.println (arr[i]);
     }
}
public static int [] mergeTwoSortedArray(int []a, int []b)
{  int [] ans = new int [a.length + b.length];
   int i = 0; -> Array 1 pe travel pe karne ke liye pointer
   int j = 0; -> 2
   int k = 0; -> Ans
   while (i < a.length && j < b.length) -> Jab tak Array 1 ya Array 2 me koi sa pura na ho jaye tab tak chalega!
   { if (a[i] < b[j])
       {
          ans[k] = a[i];
          i++;
          k++;
       }else{
          ans[k] = b[j];
          j++;
          k++;
       }
   }
   if(i == a.length) {
     while (j < b.length) {
       ans[k] = b[j];
       j++;
       k++;
   }}
 return ans;  }
```

- Time complexity : O(n)
- Space complexity : O(n)

→ Agar Array 1 ki ith value choti hui Array 2 ki jth value se toh vo Array Ans ke kth index pe jayegi! Vice-versa

→ Agar Array 1 pura traversal ho chuka ho toh Array 2 ke bachey huay elements ko Ans Array me kth index increase karty huay Daldengy!