logic - Jenny's L

Heapsort :- इसमें Array एक tree form में है ऐसा Assume करते हैं। इससे पहले heap पर होना अाकार program समझना था इसका।

Step 1 → build maxheap from array ← ये program heap में किया है & its TC is O(n)

Step 2 → now delete the root element one by one & it takes O(n log n) TC so finally TC is (n log n).

→ public class Test
{
static void A(int arr[], int n)
{
int index = (n/2)-1; // index of last non leaf node यहाँ से maxheap बनाना start करेंगे।

```
for (int i = index ; i>=0; i--)
{
    buildMaxHeap(arr, i, n);
}
for (int i = n-1 ; i>=0 ; i--)
{
    deleteMaxHeap ( arr, 0, i )
}
}
static void buildMaxHeap (int arr[], int index, int length)
{
int largest = index;
int left = 2*index + 1;    left in right child of root
int right = 2*index + 2;

if (left < length && arr[left] > arr[largest])
    largest = left;

if (right < length && arr[right] > arr[largest])
    largest = right;

if ( largest != index)
{
    // now swap करेंगे !
    int temp = arr[index];
    arr[index] = arr[largest];
    arr[largest] = temp;
    buildMaxHeap (arr, largest, length);
}
```

यहाँ फिर नीचे end तक check करेंगे।

```
{ static void deleteFromHeap(int arr[], int root,
                             int last)
    int temp = arr[root];
    arr[root] = arr[last];
    arr[last] = temp;
    buildMaxHeap(arr, 0, last-1);
}
public static void main(String args[])
{
    int arr[] = {1, 5, 9, 2, 3};
    print(arr, arr.length);
    print(Arrays.toString(arr));
}
```

Op: [2, 3, 5, 7, 9]