```java
private static void sortArr(int[] v) {
    int n = v.length, size = n;

    for (int i = n / 2; i >= 0; i--)
        heapPropertyAtNode(v, i, n);
    while (size-- > 0) {
        int temp = v[0];
        v[0] = v[size];
        v[size] = temp;
        heapPropertyAtNode(v, ind:0, size);
    }
}
```

Get Max heap.

Heapify ①

chk.

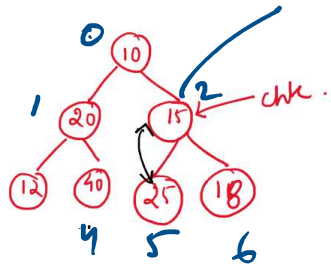
large → store ind of max-value.

$r = 2*i + 2 = 2*2+2 = 6$

$i=2,\ l = 2*i+1 = 2*2+1 = 5.$

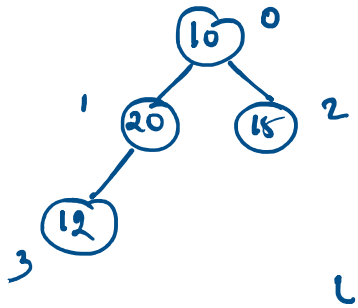
$n = 7$

$n/2 = 7/2 = 3.5 = 3$



① Start from last non-leaf node.

10 | 20 . . . . .



$n=4$

$i=1$

$l=3$

$r = 2*i+2 = 4.$

$l<n$

$r<n.$

Searching — FIND. — Linear Search → ek ek element check karo.

Binary Search.

① Sorted order.

② Search space gets reduced to half.