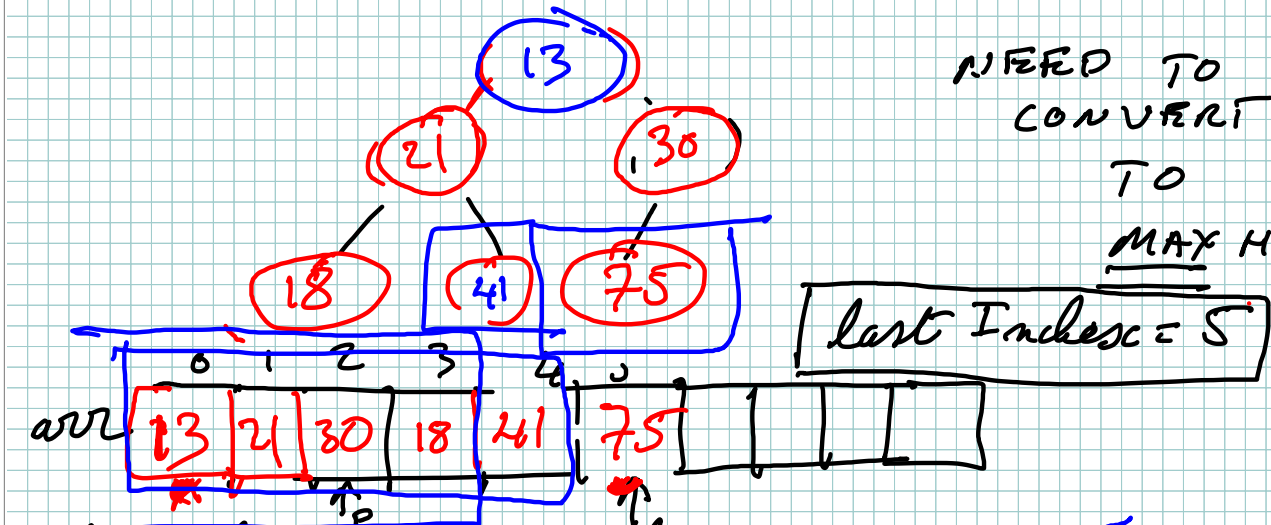


NEED TO  
CONVERT  
TO  
MAX HEAP



```

heapify() {
    int last = last Index
    int parent = (last - 1) / 2
    while (parent >= 0) {
        sift Down (parent)
        parent = parent - 1
    }
}

```

last = 5  
parent = -1

node = 4  
lc = 9  
rc = 10  
swap = 4

```

sift Down (int node) {
    while (node < last Index) {
        lc = (2 * node) + 1
        rc = (2 * node) + 2
        swap = node
        if ((lc < last Index) && (arr[node] < arr[lc])) {
            swap = lc
        }
        if ((rc < last Index) && (arr[swap] < arr[rc])) {
            swap = rc
        }
    }
}

```

```
if (swap = node) {  
    return  
} else {  
    temp = arr[node]  
    arr[node] = arr[swap]  
    arr[swap] = temp  
    node = swap  
} // end of if  
} // end of while  
} // end of siftDown.
```

```
heapSort ( ) {  
    while (last Index >= 1) {
```

$O(n \log n)$

```
        temp = arr[0]  
        arr[0] = arr[last Index]  
        arr[last Index] = temp.  
        last Index --  
        heapify ()
```

```
    }  
}
```

<

