

Heap Sort Example-

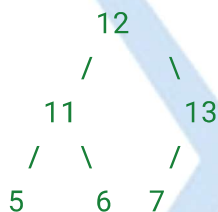
Sort the array:

[12, 11, 13, 5, 6, 7]

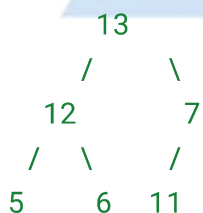
◆ Step 1: Build a Max Heap

We first convert the array into a Max Heap.

Initial array (as a binary tree):



After building a Max Heap:



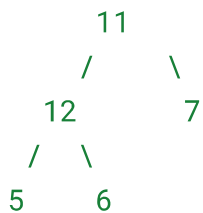
Array representation → [13, 12, 7, 5, 6, 11]

◆ Step 2: Extract the Maximum (Root)

- Swap root (13) with last element (11).
- Heap size reduces to $n-1 = 5$.

- New array → [11, 12, 7, 5, 6, 13]

Now **heapify** the reduced heap.



After heapify:

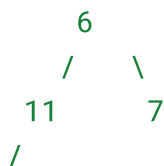


Array → [12, 11, 7, 5, 6, 13]

◆ Step 3: Extract Next Maximum

- Swap root (12) with last element in heap (6).
- Heap size = 4.
- Array → [6, 11, 7, 5, 12, 13]

Heapify again:



5

After heapify:

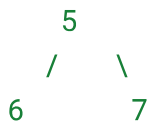


Array → [11, 6, 7, 5, 12, 13]

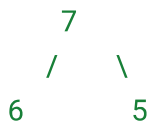
◆ Step 4: Extract Next Maximum

- Swap root (11) with last element in heap (5).
- Heap size = 3.
- Array → [5, 6, 7, 11, 12, 13]

Heapify:



After heapify:



Array → [7, 6, 5, 11, 12, 13]

◆ Step 5: Extract Next Maximum

- Swap root (7) with last element in heap (5).
- Heap size = 2.
- Array → [5, 6, 7, 11, 12, 13]

Heapify:

5
/
6

After heapify:

6
/
5

Array → [6, 5, 7, 11, 12, 13]

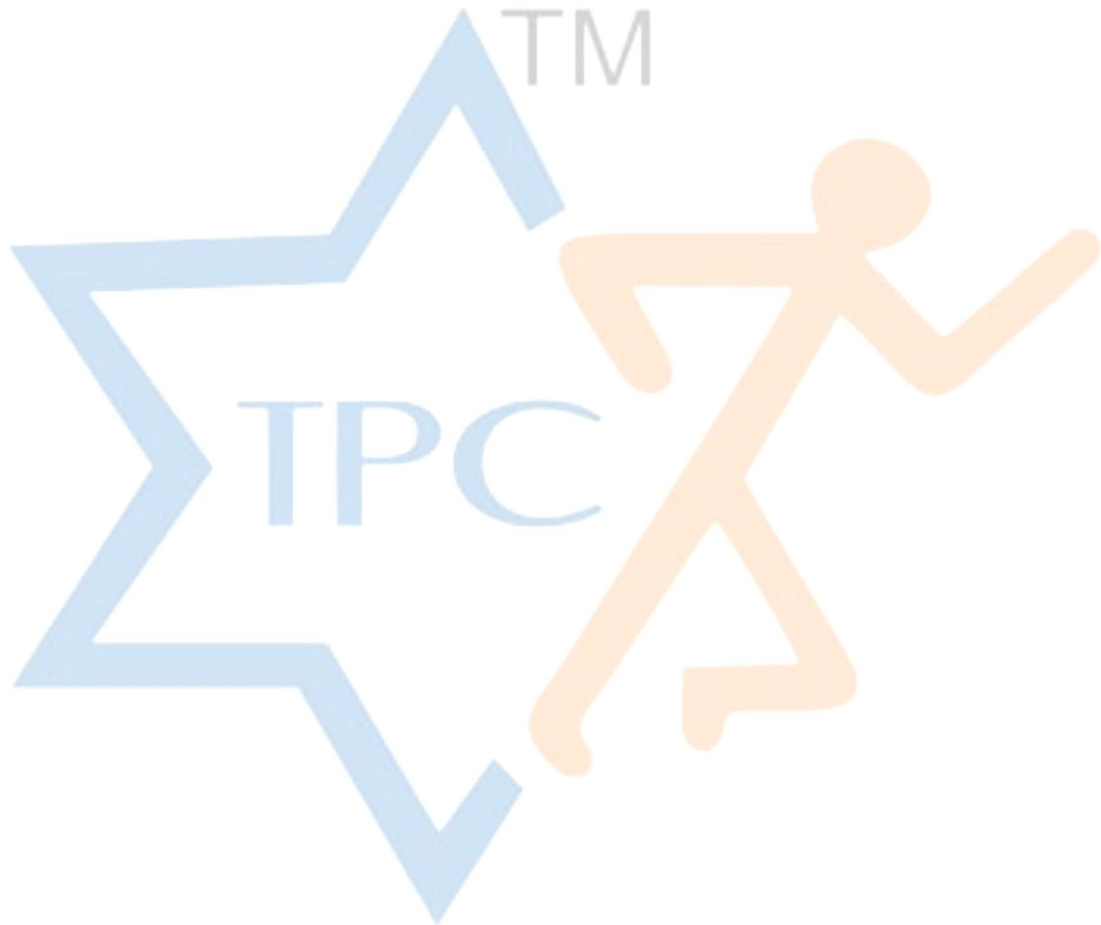
◆ Step 6: Extract Next Maximum

- Swap root (6) with last element in heap (5).
- Heap size = 1.
- Array → [5, 6, 7, 11, 12, 13]

Now only one element remains, so the heap is sorted.

Final Sorted Array:

[5, 6, 7, 11, 12, 13]



www.tpcglobal.in