Course title : **CSE2001**

Course title : **Data Structures and Algorithms**

Module : **4**

Topic : **4**

Heap Sort

Objectives

This session will give the knowledge about

- Heap sort

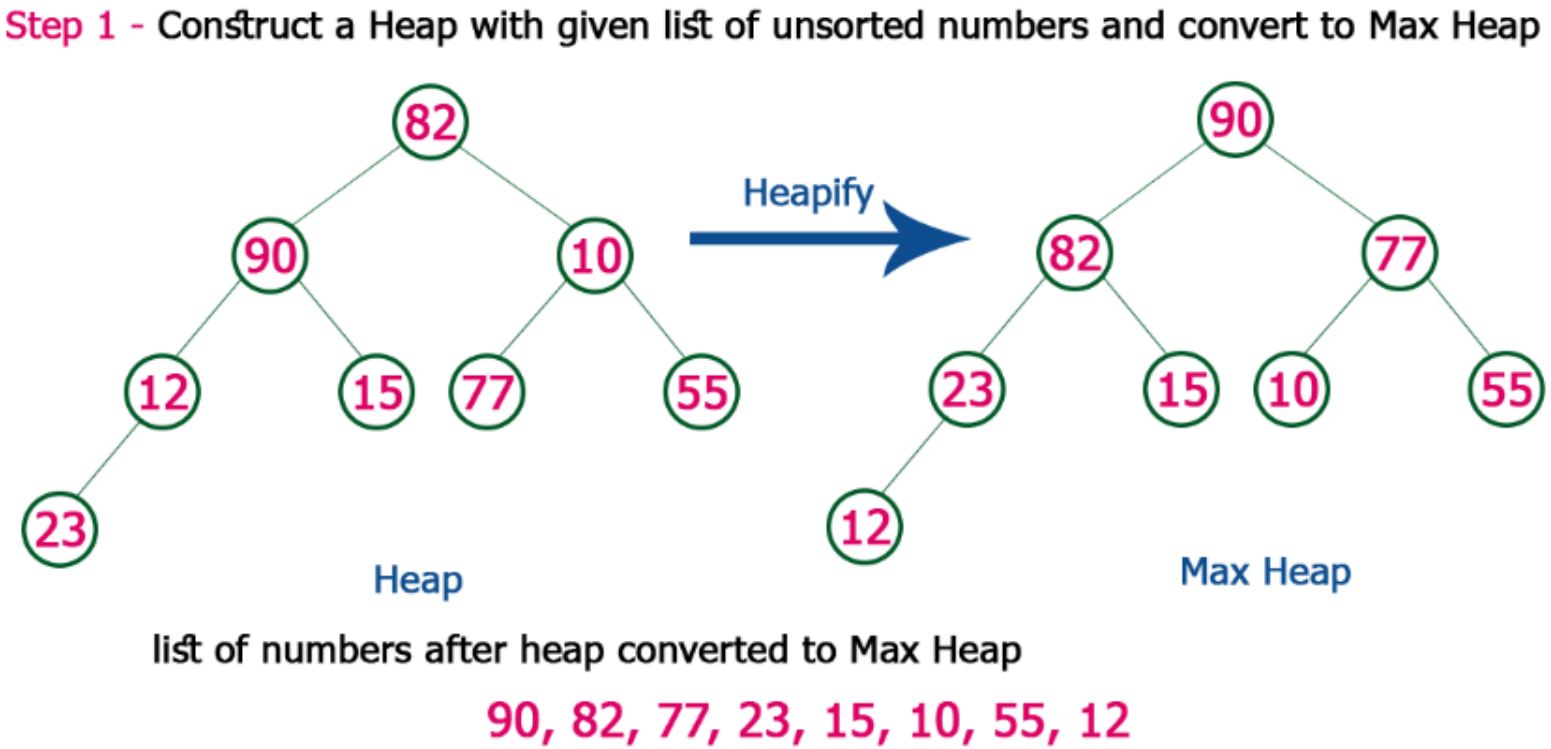
Heap Sort

- Heapsort algorithm uses one of the tree concepts called Heap Tree. In this sorting algorithm, we use Max Heap to arrange list of elements in Descending order and Min Heap to arrange list elements in Ascending order.

- Step by Step Process
  - Step 1 - Construct a Binary Tree with given list of Elements.
  - Step 2 - Transform the Binary Tree into Min or Max Heap.
  - Step 3 - Delete the root element from Min Heap using Heapify method.
  - Step 4 - Put the deleted element into the Sorted list.
  - Step 5 - Repeat the same until Min Heap becomes empty.
  - Step 6 - Display the sorted list.

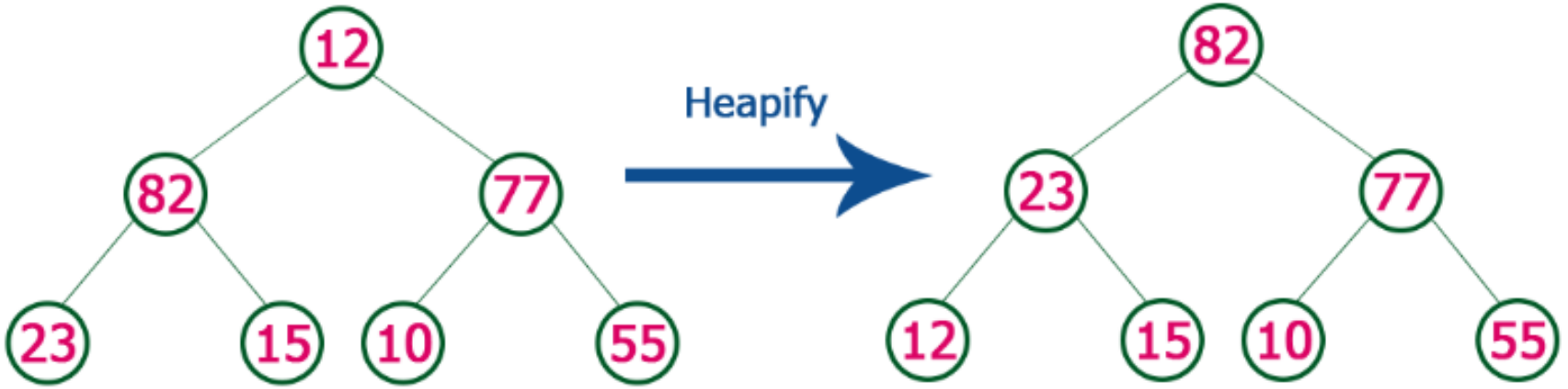
Heap Sort – Example (Max Heap)

- Consider list = 82 90 10 12 15 77 55 23



Heap Sort – Example (Max Heap)

Step 2 - Delete root (**90**) from the Max Heap. To delete root node it needs to be swapped with last node (**12**). After delete tree needs to be heapify to make it Max Heap.



Heap after 90 deleted

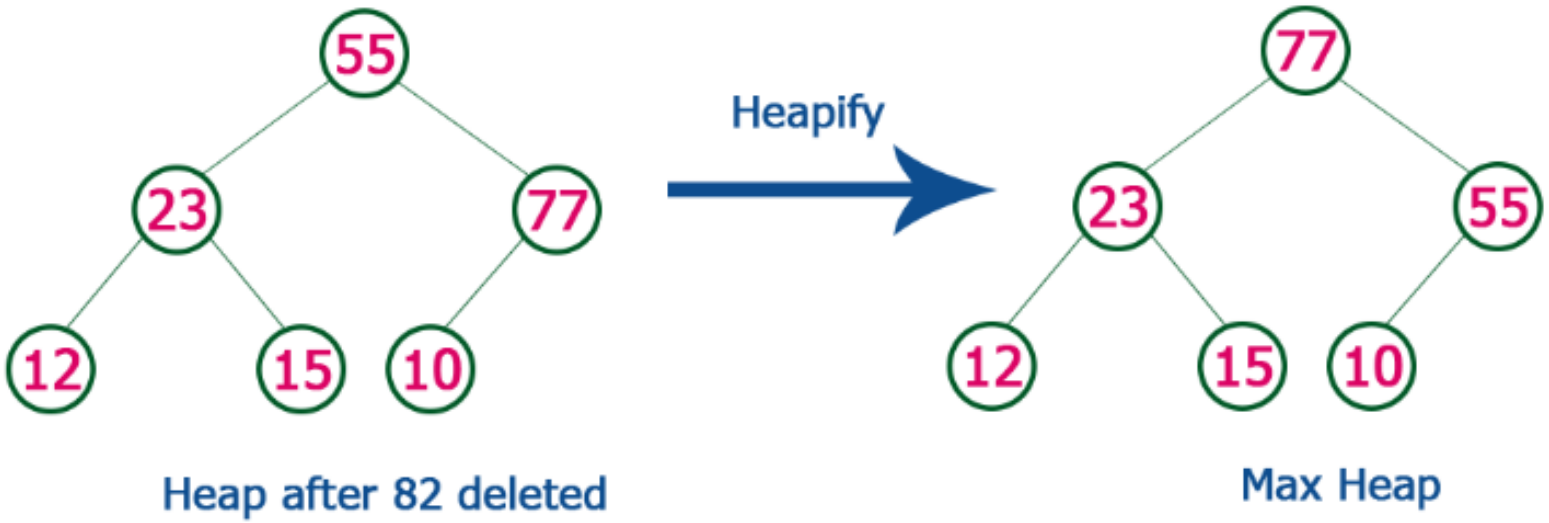
Max Heap

list of numbers after swapping 90 with 12.

12, 82, 77, 23, 15, 10, 55, **90**

Heap Sort – Example (Max Heap)

Step 3 - Delete root (**82**) from the Max Heap. To delete root node it needs to be swapped with last node (**55**). After delete tree needs to be heapify to make it Max Heap.



Heap after 82 deleted

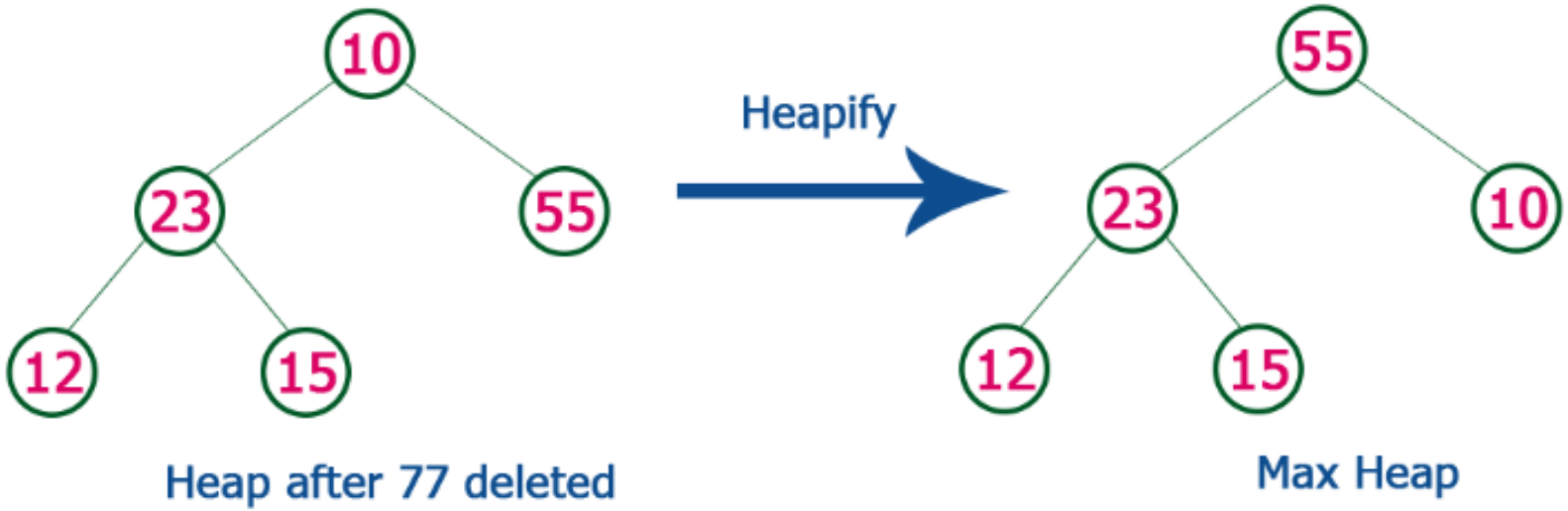
Max Heap

list of numbers after swapping 82 with 55.

12, 55, 77, 23, 15, 10, **82**, **90**

Heap Sort – Example (Max Heap)

Step 4 - Delete root (**77**) from the Max Heap. To delete root node it needs to be swapped with last node (**10**). After delete tree needs to be heapify to make it Max Heap.

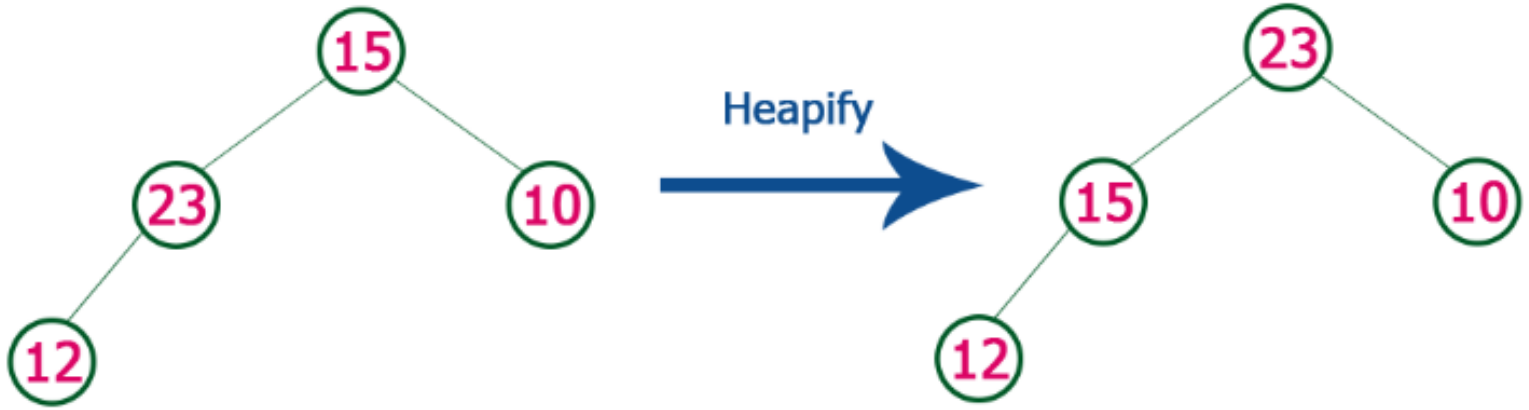


list of numbers after swapping 77 with 10.

12, 55, 10, 23, 15, **77**, **82**, **90**

Heap Sort – Example (Max Heap)

Step 5 - Delete root (**55**) from the Max Heap. To delete root node it needs to be swapped with last node (**15**). After delete tree needs to be heapify to make it Max Heap.



Heap after 55 deleted

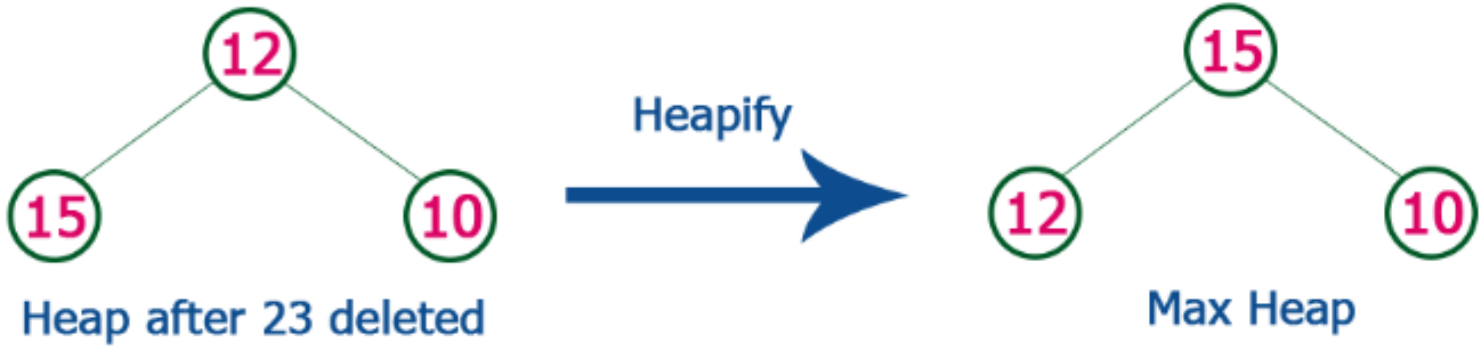
Max Heap

list of numbers after swapping 55 with 15.

12, 15, 10, 23, **55, 77, 82, 90**

Heap Sort – Example (Max Heap)

Step 6 - Delete root (**23**) from the Max Heap. To delete root node it needs to be swapped with last node (**12**). After delete tree needs to be heapify to make it Max Heap.

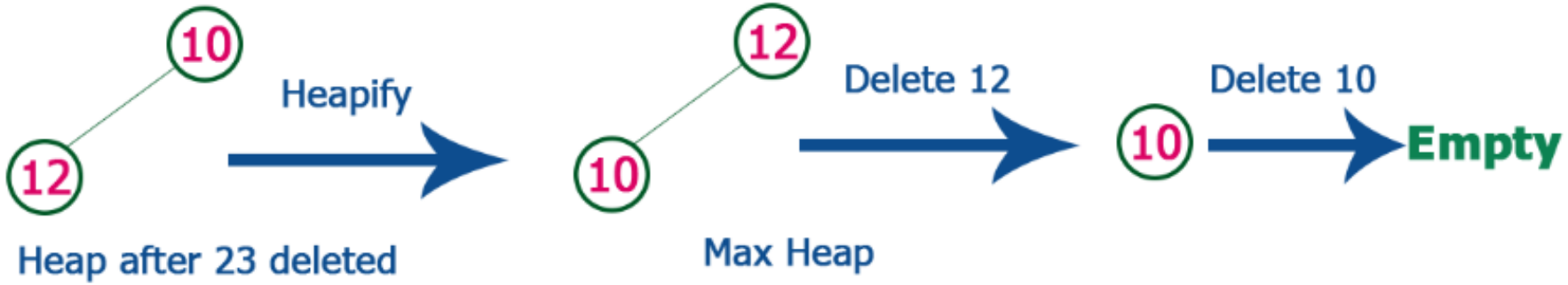


list of numbers after swapping 23 with 12.

12, 15, 10, **23, 55, 77, 82, 90**

Heap Sort – Example (Max Heap)

Step 7 - Delete root (**15**) from the Max Heap. To delete root node it needs to be swapped with last node (**10**). After delete tree needs to be heapify to make it Max Heap.



list of numbers after Deleting 15, 12 & 10 from the Max Heap.

10, 12, 15, 23, 55, 77, 82, 90

Whenever Max Heap becomes Empty, the list get sorted in Ascending order

Heap Sort – Min Heap Algorithm

```
static void heapsort(int arr[]) {
        int n = arr.length;
        for (int i = n / 2 - 1; i >= 0; i--) {
            heapify(arr, n, i);
        }
        for (int i = n - 1; i >= 0; i--) {
            int temp = arr[0];
            arr[0] = arr[i];
            arr[i] = temp;
            heapify(arr, i, 0);
        }
}
```

```
static void heapify(int arr[], int n, int i) {
        int root = i;
        int left = 2 * i + 1;
        int right = 2 * i + 2;
        if (left < n && arr[left] > arr[root])
                root = left;
        if (right < n && arr[right] > arr[root])
                root = right;
        if (root != i) {
            int swap = arr[i];
            arr[i] = arr[root];
            arr[root] = swap;
            heapify(arr, n, root);
        }
}
```

Complexity Analysis

Algorithm	Time Complexity		
	Best	Average	Worst
Selection Sort	Ω(n^2)	θ(n^2)	O(n^2)
Bubble Sort	Ω(n)	θ(n^2)	O(n^2)
Insertion Sort	Ω(n)	θ(n^2)	O(n^2)
Heap Sort	Ω(n log(n))	θ(n log(n))	O(n log(n))
Quick Sort	Ω(n log(n))	θ(n log(n))	O(n^2)
Merge Sort	Ω(n log(n))	θ(n log(n))	O(n log(n))
Bucket Sort	Ω(n+k)	θ(n+k)	O(n^2)
Radix Sort	Ω(nk)	θ(nk)	O(nk)

Summary

At the of this session we learned about

- Heap Sort