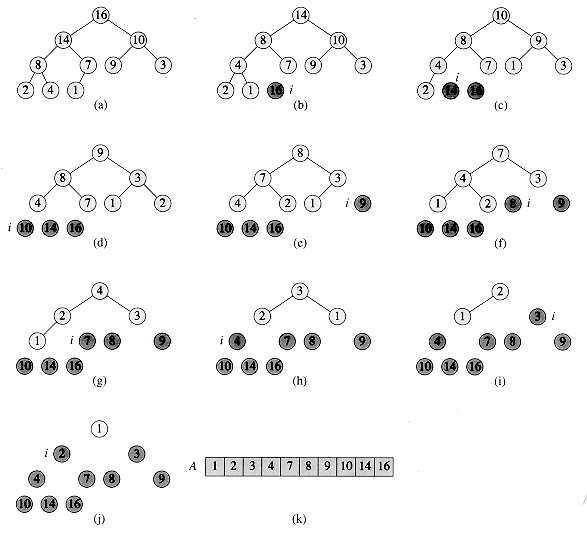
**Heap sort sorting technique**

Merge sort is an sorting technique where an unsorted array is converted into a binary tree and then sorted accordingly. There are two types of heaps(i.e: minimum and maximum heap. Minimum heap is used to sort elements from smaller to bigger and maximum heap used to vice versa of the same.

****

**Working of heap sort:**

1. **Convert the unsorted array into the heap as per the requirement.**
2. **We should heapify the whole heap.**

**Conditions to heapify:**

**In minimum heap the parent node of the root should be always smaller then the child node.**

**In maximium heap the parent node of the root should be always greater then the child node.**

1. **Then swap the first element to the last element.**
2. **Then repeat the step 1 to step3 till the sorted array is obtained..**

**Function to heapify (pseudo code):**

MaxHeapify(A, i)

l = left(i)

r = right(i)

if l <= heap-size[A] and A[l] > A[i]

then largest = l

else largest = i

if r <= heap-size[A] and A[r] > A[largest]

then largest = r

if largest != i

then swap A[i] with A[largest]

MaxHeapify(A, largest)

end func

**Function to build heap (pseudo code)**

BuildMaxHeap(A)

heap-size[A] = length[A]

for i = |length[A]/2| downto 1

do MaxHeapify(A, i)

end func

**Function to sort the heap**

HeapSort(A)

BuildMaxHeap(A)

for i = length[A] downto 2

do swap A[1] with A[i]

heap-size[A] = heap-size[A] – 1

MaxHeapify(A, 1)

end func