

HEAP SORT

In this Algorithm we first build the heap using the given elements.

We create a MaxHeap to sort the elements in ascending order.

Once the heap is created we swap the root node with the last node and delete the last node from the heap.

```
void heapSort(int arr[], int n)
{
    for(int i = n/2 - 1; i >= 0; i--)
        heapify(arr, n, i);
    for(int i = n - 1; i >= 0; i--)
    {
        swap(arr[0], arr[i]);
        heapify(arr, i, 0);
    }
}
```

```
Heapsort (A as array)
BuildMaxHeap(A)
for i = n to 1
    swap (A[1], A[i])
    n = n - 1
    Heapify (A, 1)
```

```
BuildMaxHeap (A as array)
```

$O(n \log n)$

build-max-heap : $O(n)$

heapify : $O(\log n)$, called $n-1$ times

```
n = elements_in(A)
for i = floor (n/2) to 1
    Heapify (A,i)
```

Heapify (A as array, i as int)

```
left = 2i
```

```
right = 2i+1
```

```
if (left <= n) and (A[left] > A[i])
```

```
    max = left
```

```
else
```

```
    max = i
```

```
if (right <= n) and (A[right] > A[max])
```

```
    max = right
```

```
if (max != i)
```

```
    swap (A[i], A[max])
```

```
    Heapify (A, max)
```

