

CS218 - Data Structures  
FAST NUCES Peshawar Campus  
Dr. Nauman (recluze.net)

October 8, 2019

## 1 Trees

Raster images of the notebook 14-heap.

### Heap Data Structure and Heap Sort

#### Heap Data Structure - MaxHeap and MinHeap ¶

```
In [ ]: n = 8
        for i in reversed( range( n ) ):
            print(i)
```

```
In [ ]: def heapify(lst, n, root):
        """Heapify the root element of lst which has n elements in total."""

        largest = root
        l = 2*root + 1
        r = 2*root + 2

        if l < n and lst[l] > lst[largest]:
            largest = l;

        if r < n and lst[r] > lst[largest]:
            largest = r

        if largest != root:
            lst[root], lst[largest] = lst[largest], lst[root]

            heapify(lst, n, largest)
```

```
In [ ]: def build_heap(lst):
        """Construct heap from any list by repeatedly applying heapify from bottom non-leaf to top."""

        n = len(lst)

        for i in reversed( range( n // 2 ) ):    # rest must be leaves, so, no need to heapify them
            # print("Heapifying: ", lst[i])
            heapify(lst, n, i)
```

```
In [ ]: heap = [100, 5, 3, 2, 8, 15, 6, 102]
```

```
In [ ]: print(heap)
        build_heap(heap)
        print(heap)
```

## Heap Sort

```
In [ ]: def heap_sort(lst):
        """Perform Heap Sort by building a Max Heap from the list and then repeatedly retrieving the max and heapifying the rest."""
        n = len(lst)

        # Build max heap
        build_heap(lst)

        for i in reversed(range(n)):
            # swap
            lst[i], lst[0] = lst[0], lst[i]

            #heapify root element again but only until the ith element
            heapify(lst, i, 0)
```

```
In [ ]: heap = [100, 5, 3, 2, 8, 15, 6, 102]
        print(heap)
        heap_sort(heap)
        print(heap)
```