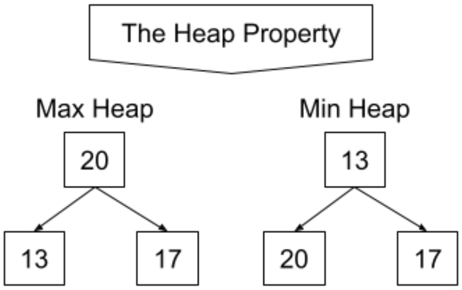
Topic 8 - Heaps

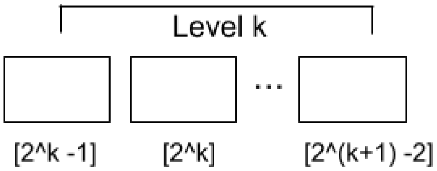
Heaps: Introduction

- A heap is a **tree data structure**
- **The heap property.** This property states what the value of the parent node must be in relation to the value of its children
- **The shape property:**
    - All levels full (**perfect triangle**), except maybe the last one
    - The last level must be filled from left to right (**left-aligned rectangle**)
- Other type of heap trees:
    - **Ternary Heap**
    - **Quaternary Heap**
    - **N-ary Heap**



Heaps: Implementation

- Arrays is the most common way to implement Heaps
- This way of representing a heap is called **Implicit Representation**



NODE	PARENT	LEFT CHILD	RIGHT CHILD
[0]	---	[1]	[2]
[1]	[0]	[3]	[4]
[2]	[0]	[5]	[6]
[3]	[1]	[7]	[8]
[4]	[1]	[9]	[10]
[5]	[2]	[11]	[12]
[K]	$FLOOR[(K - 1)/2]$	$2K + 1$	$2K + 2$

```
function PARENT( k )
    return  FLOOR((k-1) / 2)
end function
function LEFT ( k )
    return 2*k+1
end function
function RIGHT ( k )
    return 2*k+2
end function
```

Heaps: Insert (element by element)

MAX HEAP

```
function INSERT (heap, k)
    pos = heap_size
    heap[pos]=k
    heap_size = heap_size + 1
    // check that is a heap
    while ( pos > 0 && heap[PARENT( pos )] < heap[pos] )
        SWAP(heap[PARENT( pos )], heap[pos])
        pos = parent( pos )  // position of the parent
    end while
end function
```

Heaps:deletion (extract maximum)

If this is applied you sort the elements in descending order

```
T(N) : Θ(log(N))
function EXTRACT-MAX( heap )
    max = heap[0]
    heap[0] = heap[heap_size-1] // copy last element of array
    heap_size = heap_size -1
    MAX-HEAPIFY( heap, 0 )
    return max
end function

root : index number EG: 0  T(N) : Θ(log(N))
function MAX-HEAPIFY( heap,root )
    largest = INDEX_LARGETS_NODE( root )// index of the largest
value
    if(largest != root)
        SWAP( heap[largest], heap[root] )
        MAX-HEAPIFY (heap, largest)
    end if
end function
```

Heaps: Build in place

- **Copy the elements of a binary tree into a heap**, elements might satisfy the shape property but not the heap property
- Methods to copy the elements:
    - **Out of place**: creates a new array and using the INSERT Method in each of the elements of the binary tree
    - **In Place:** Move the elements of the array without creating another array

```
A: Array   T(N) : Θ(N*log(N))
function BUILD-MAX-HEAP (A)
    heap_size = A.length
    for FLOOR ( heap_size/2 ) > j >= 0
        MAX-HEAPIFY(A,j)
    end for
function
```

Heapsort

- Unsorted Array
- Build HEAP in Place
- Extract MAX and copy at the end
- repeat until the array is sorted

```
A: Array   T(N) : Θ(N*log(N))
function HEAPSORT(A)
    BUILD_MAX_HEAP(A)
    while heap_size > 0
        i = heap_size-1
        A[i] = EXTRACT_MAX(A)
    end while
    return array
end function
```

Note: heaps are also used to implement priority queues