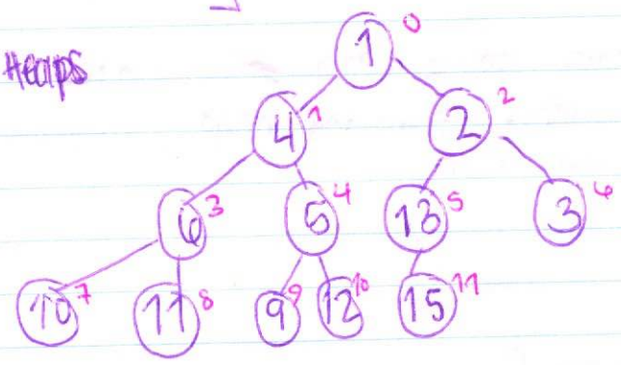
Nov 17
COMP 2402

[Heaps & HeapSort]

Binary Heaps



0	$1 = 2^0$
1	$2 \leq 2^1$
2	$4 \leq 2^2$
3	$8 \leq 2^3$

Eytzinger's method

1	4	2	6	5	13	3	10	11	9	12	15
0	1	2	3	4	5	6	7	8	9	10	11

$left(i) = 2i + 1$ $\quad right(i) = 2i + 2$

$parent(i) = \left\lfloor \frac{(i-1)}{2} \right\rfloor$

$2^h \leq n$

$h \leq \log(n)$

```
add(x)
  resize if full    // O(1) amortized
  a[n++] = x        // O(1)
  "bubble up"       // O(h)
```

} total $O(\log n)$ amortized

```
remove()
  ty = a[0]
  a[0] = a[n-1]
  n--
  heapify           // O(log n)
  if (too empty) resize();
  return ty;
```

} $O(1)$

Heap Sort

(1) add n items to a heap // nlogn
(2) remove n items from the heap
(3) reverse the array

Another way:

(1) Build the heap from the bottom up.
(2) remo

heapify (leaf) = 0 swaps
heapify (leaf.parent) ≤ 1 swap
≤ 2 swaps
⋮
≤ h

$$\leq \sum_{i=1}^{\log n} (i-1) \cdot n/2^i$$

(i-1): # of swaps for a layer
$n/2^i$: # of nodes at layer i

$$\leq \sum_{i=1}^{\infty} i/2^i = n \cdot (1 \cdot 1/2 + 2 \cdot 1/4 + 3 \cdot 1/8 + \dots)$$

$$= n(\tfrac{1}{2} + \tfrac{1}{4} + \tfrac{1}{8} \dots) + (\tfrac{1}{4} + \tfrac{1}{8} + \dots)$$

$$= n \cdot (1 + \tfrac{1}{2} + \tfrac{1}{4} + \tfrac{1}{8} \dots$$

$$= 2n$$