

Heaps & Heapsort

Tuesday, November 3, 2015

11:37 AM

Quiz 04

Using Mergesort, sort the following functions in increasing order (resolve conflicts arbitrarily)

~~$n^2 \lg n$~~ , ~~n^3~~ , ~~$\lg n$~~ , 2^n , ~~$n \lg n$~~ , ~~n^2~~ , ~~n^n~~ , ~~n~~

Solution:

$\lg n$	n	$n \lg n$	n^2	$n^2 \lg n$	n^3	2^n	n^n
---------	-----	-----------	-------	-------------	-------	-------	-------

$n^2 \lg n$	n^3	$\lg n$	2^n	$n \lg n$	n^2	n^n	n
-------------	-------	---------	-------	-----------	-------	-------	-----

$n^2 \lg n$	n^3	$\lg n$	2^n
-------------	-------	---------	-------

$n \lg n$	n^2	n^n	n
-----------	-------	-------	-----

$n^2 \lg n$	n^3
-------------	-------

$\lg n$	2^n
---------	-------

$n \lg n$	n^2
-----------	-------

n^n	n
-------	-----

$n^2 \lg n$ n^3

$\lg n$ 2^n

$n \lg n$ n^2

n^n n

$n^2 \lg n$	n^3
-------------	-------

$\lg n$	2^n
---------	-------

$n \lg n$	n^2
-----------	-------

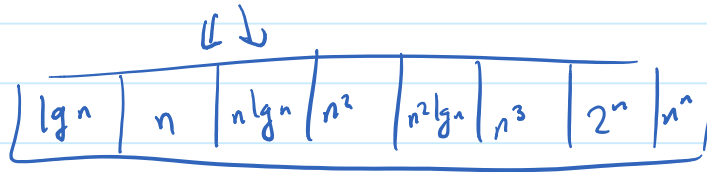
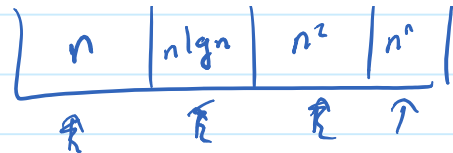
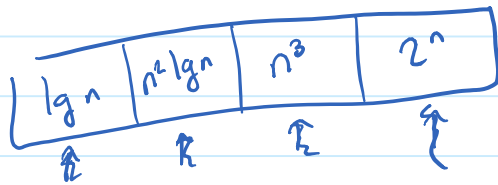
n	n^n
-----	-------

$\nearrow$ $\nearrow$ $\uparrow$ $\nearrow$ $\uparrow$

$\uparrow$ $\uparrow$ $\uparrow$ $\uparrow$

n	$n \lg n$	n^2	n^n
-----	-----------	-------	-------

n	$n \lg n$	n^2	n^n
-----	-----------	-------	-------



$$2n = O(n)$$

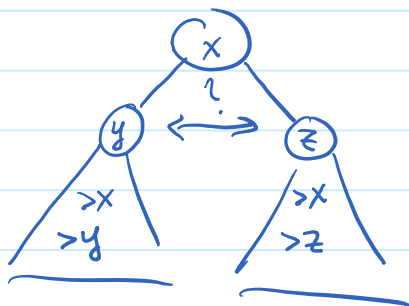
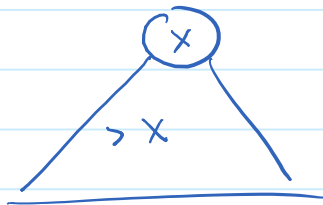
$$n = O(n)$$

$$\lg n$$

$$\log n$$

Heaps

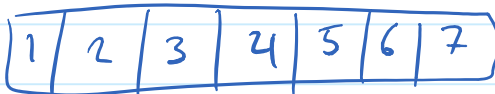
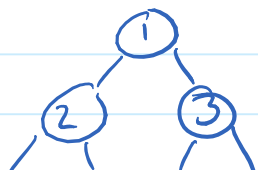
	insertion	quick	merge
best	n	$n \lg n$	$n \lg n$
avg	n^2	$n \lg n$	$n \lg n$
worst	n^2	n^2	$n \lg n$

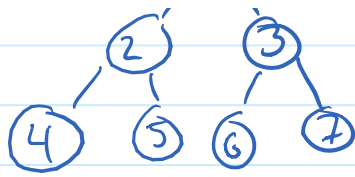


$$y > x$$

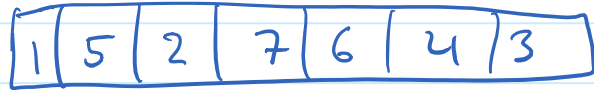
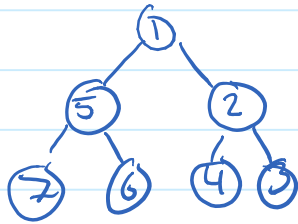
$$z > x$$

Ex 1:

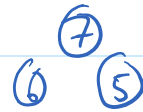
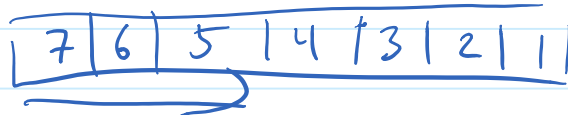




Ex 2:



heapify $O(n \lg n)$



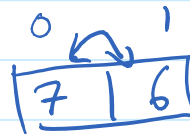
step 1



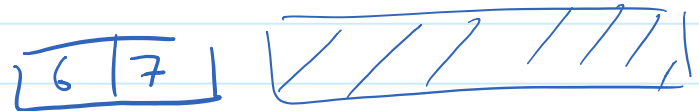
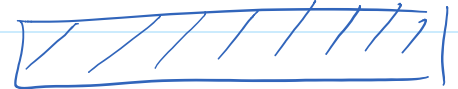
step 2



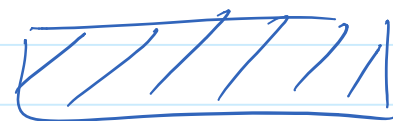
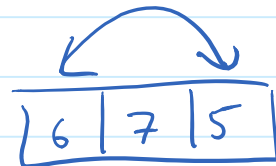
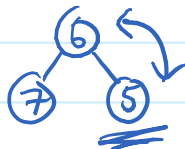
- insert the next element
- bubble that element up until the heap property is satisfied

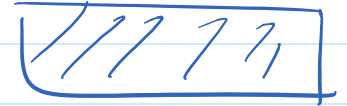
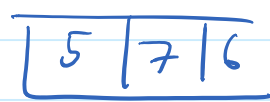
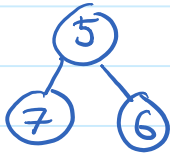


left (index): $2 * \text{index} + 1$

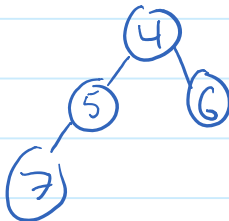
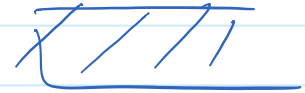
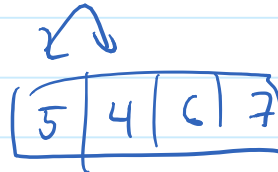
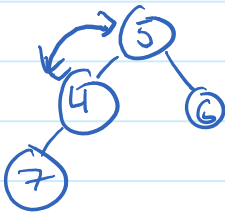
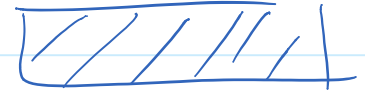
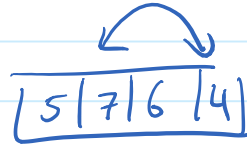
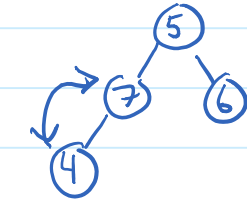


step 3

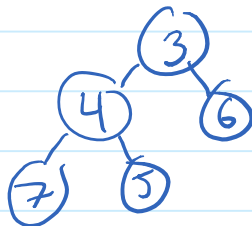
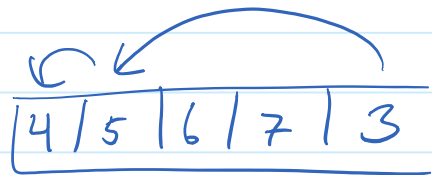
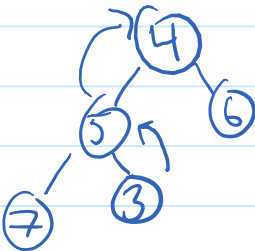




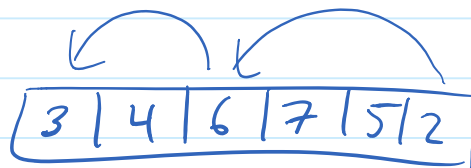
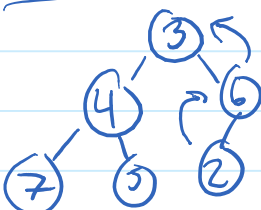
Step 4

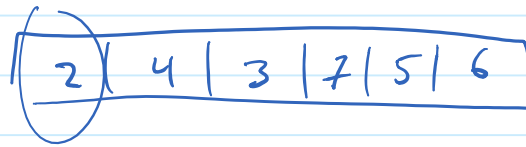
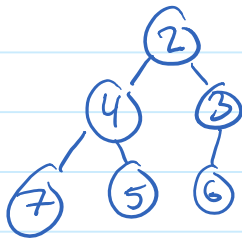


Step 5

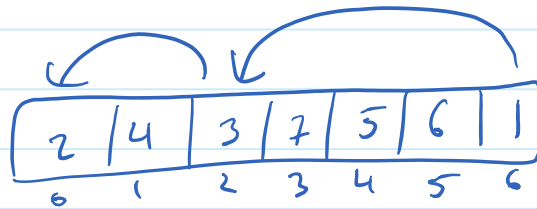
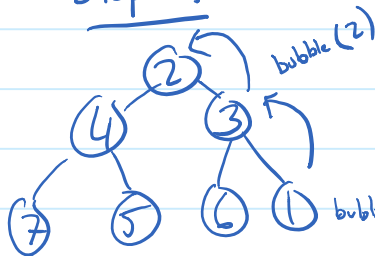


Step 6

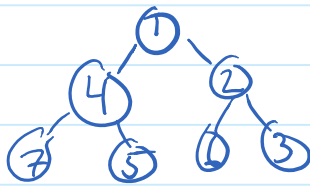




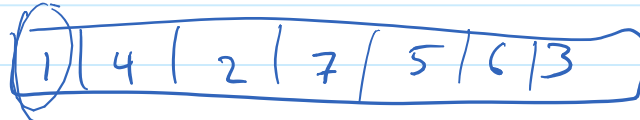
Step 7



$$\text{right}(\text{index}) = 2 * \text{index} + 2$$



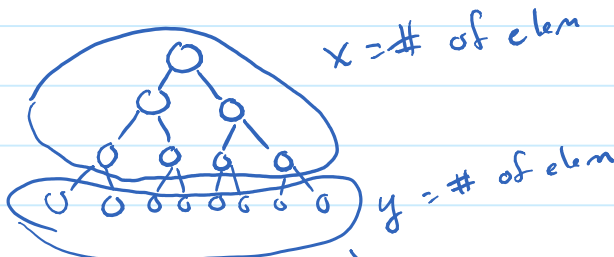
$$h = \lg n$$



$$\Omega(n) \quad h \propto \lg n$$

$$\text{Heapify} = \Theta(n \lg n)$$

There are heapify algorithms in $\Theta(n)$



$$x \sim y$$

Heap(E extends Comparable(E))

getMin() { // peek

return list.get(0)

}

```

insert (E elem) {
    list.append(elem)
    bubbleUp(list.size()-1)
}

```

```

deleteMin() {
    toReturn = getMin();
    index = 0;

```

```

    while (index is within the list) {

```

```

        if (leftE(index) > rightE(index)) {

```

```

            bubbleup(right)

```

```

            index = right(index)

```

```

        } else {

```

```

            bubbleUp(left)

```

```

            index = left(index)

```

```

        }

```

```

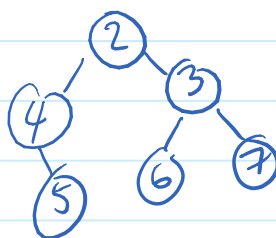
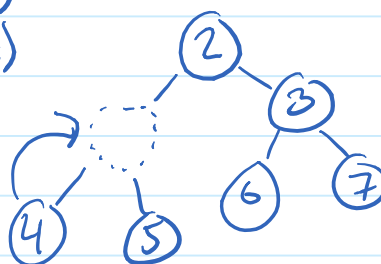
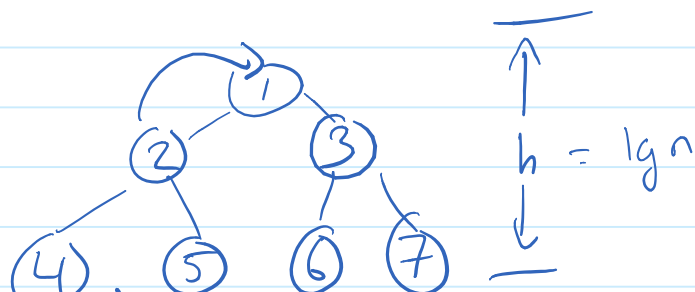
    }

```

```

}

```

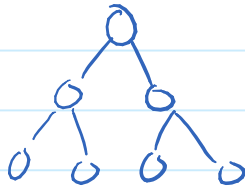


deleteMin = $\lg n$

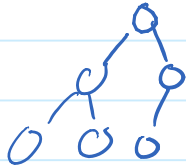
heapSort

best $\Theta(n \lg n)$
 worst $\Theta(n \lg n)$
 avg $\Theta(n \lg n)$

	heap	insert	quick	merge
best	$n \lg n$	n	$n \lg n$	$n \lg n$
avg	$n \lg n$	n^2	$n \lg n$	$n \lg n$
worst	$n \lg n$	n^2	$n \lg n$	$n \lg n$



$\lg n$



$\lg n$

$n \lg n$