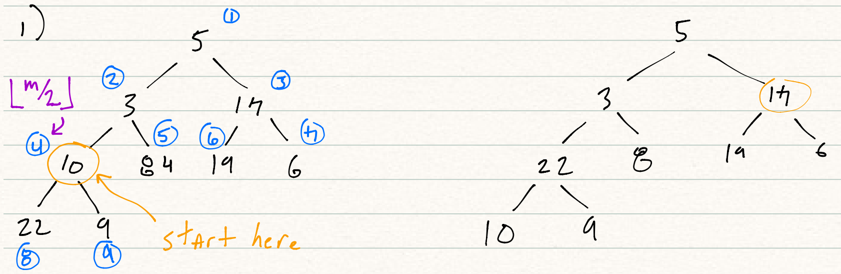
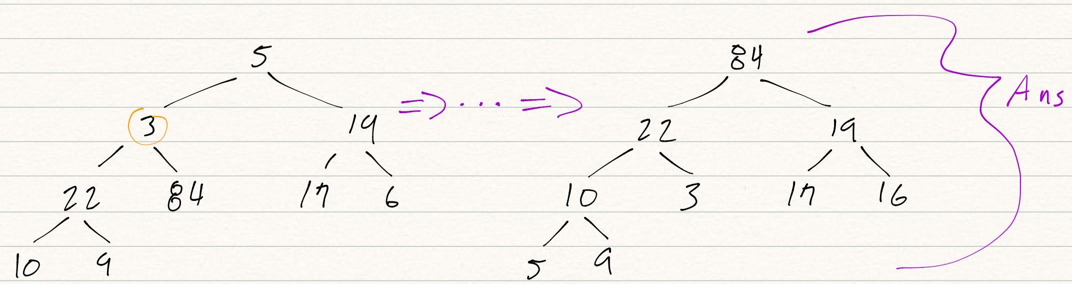
- Mid-term
  - 10? Q's
  - no Random Variables Q
  - Ch's: 6 (Heaps, Heapsort, Priority Queues), Ch. 7 (Quicksort),
    8 (Lower bound on sorting, Counting Sort, Radix sort, Bucket sort),
    22 (Breadth first search)

- Hw 03

1)





2) B/c the children are NOT max heaps => won't get largest element @ the root

3) HEAP-DELETE(A, i)

```
// mark A[i] with -∞
A[i] = -∞
MAX-HEAPIFY(A, i)
remove A[A.length]
? -> i--;
```

4) $T(n) = \Theta(n) + T(n/2) + T(n/2)$
$= 2T(n/2) + \Theta(n)$

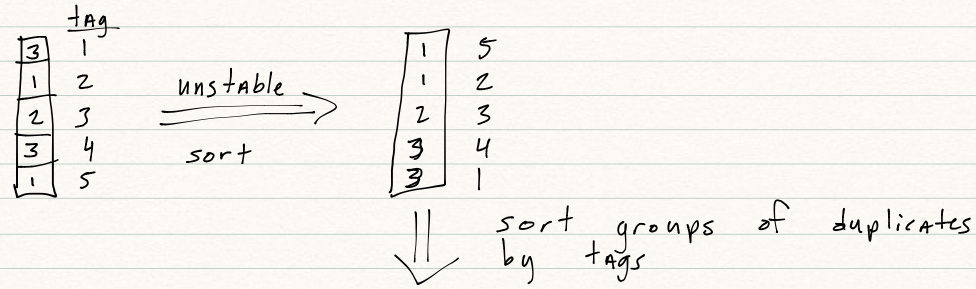
$\Rightarrow \Theta(n \lg n)$
$\Rightarrow \Omega(n \lg n)$

5) minimum depth = n-1

6) insertion sort: preserves order
heap sort: doesn't
quicksort: doesn't

merge sort: preserves order

1st Possiblity



4)

• Sample Mid-term

1) For a list of n items there are $n!$ possible orderings, that means the decision tree will have $n!$ leaves

$$c n \lg n \le \log_2(n!) \le \lg(n^n) = n \lg(n)$$

$$\log_2(n!) = \sum_{i=1}^{n} \log(i) \ge n \lg n$$

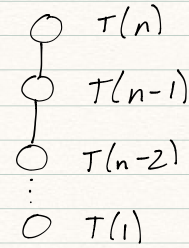
On test, can just say: "As proved in Ch.3"

2)

3)
75% of time $T(n) = T\left(\frac{3n}{4}\right) + T\left(\frac{n}{4}\right) + \Theta(n) = \Theta(n \lg(n))$

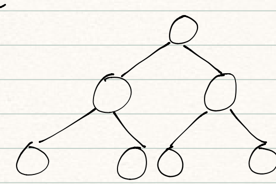
25% of time $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) = \Theta(n \lg(n))$

4) worst-case: NO BRANCHING



$\Theta(n)$ calls to the random generator

best-case: BRANCHING, FULL TREE



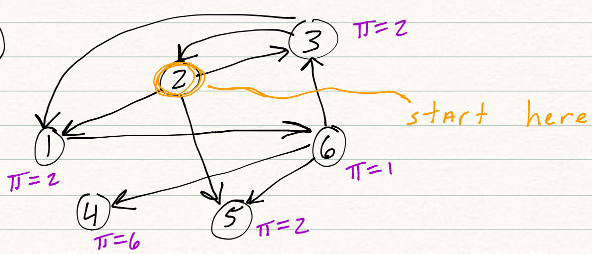
at level k, 2^k calls to partition k levels, where 2^k = n

$$\sum_{i=1}^{k} 2^k = 2^{k+1} - 1$$

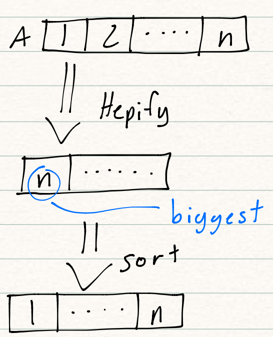
$$= n^2$$

$\Theta(n^2)$ He will send email if incorrect

5)



6)



* This is the worst-case for heapsort

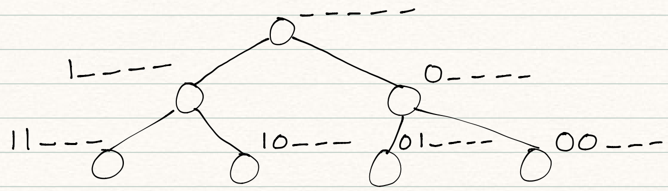
7)

MAX-Heap-Insert(A, key++) => most recent key is @ the top

LOOK @ THIS Q!!
Hinted that there would be
a stack Q

8)

n-digit binary



same amount of time as starting w/ Least Significant

space is terrible b/c have to keep track of all the
the piles