

EECS 281, Week 3: Wednesday

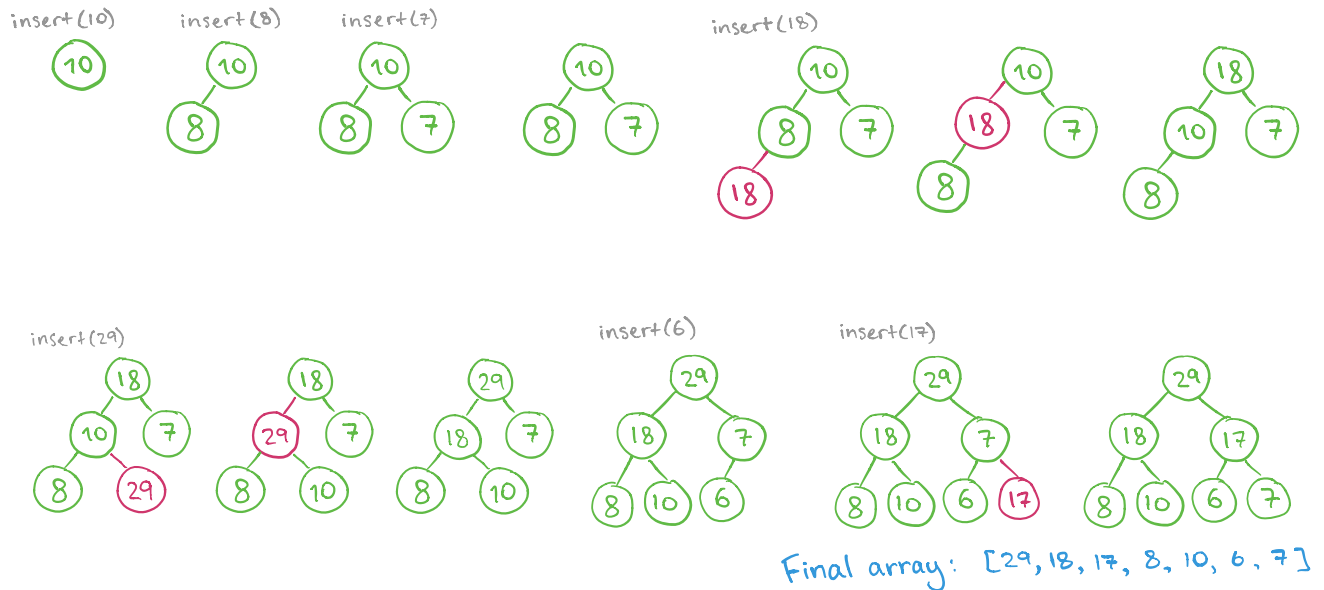
Heaps

Suppose that seven values are pushed into an empty maximum priority queue binary heap in the following order:

10, 8, 7, 18, 29, 6, 17

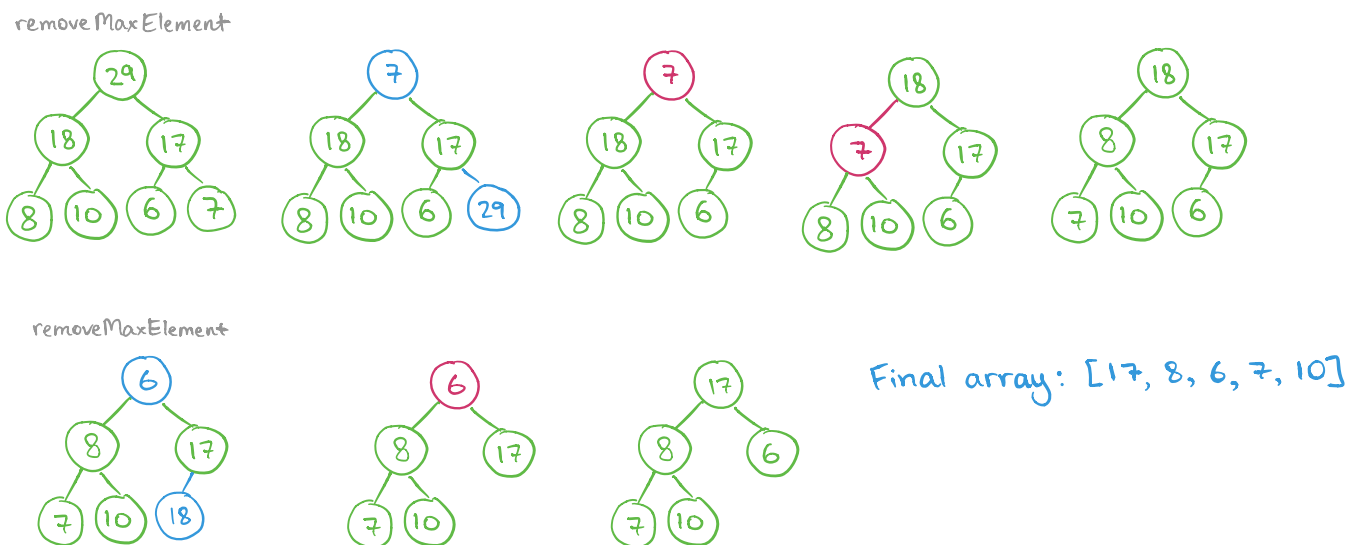
What would the array representation of the heap look like after all seven values are inserted?

Draw the tree representing the heap after each insertion.



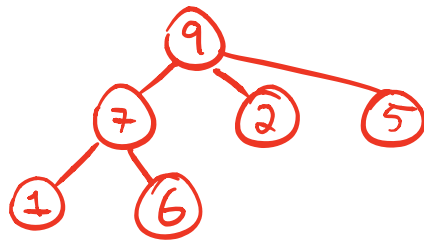
What is the array representation of the heap after calling `removeMaxElement` twice?

Draw the tree representing the heap.



(Max) Pairing Heap

Not array!

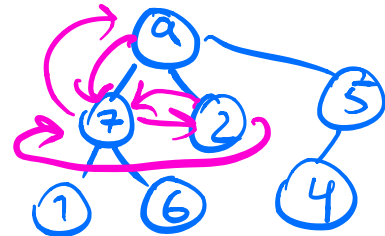
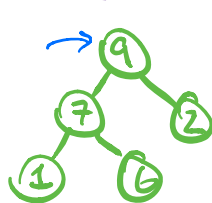


size track every time add/remove elt

maxElement

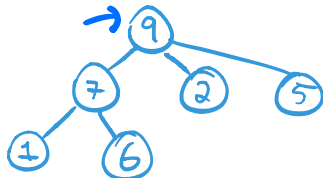
root (pointer)

$\Theta(1)$ meld

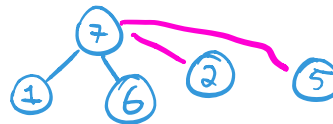


removeMaxElement

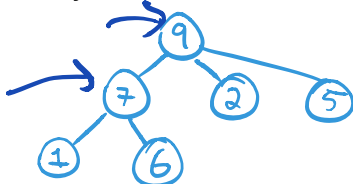
$\log(n)$



meld

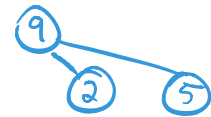
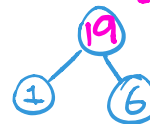


increaseKey

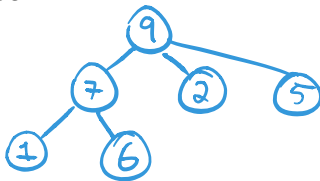


$7 \rightarrow 8 \Rightarrow$ do nothing

$7 \rightarrow 19$

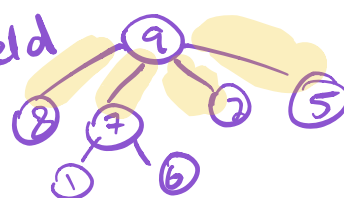


insert

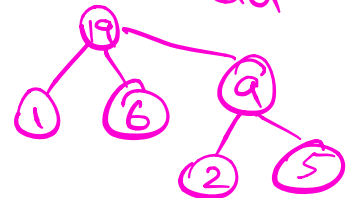


8

meld



meld



updatePriorities

detach everything

1 7 2

keep melding pairs on P PQs until 1 is left

more efficient than sorted array PQ or binary heap overall

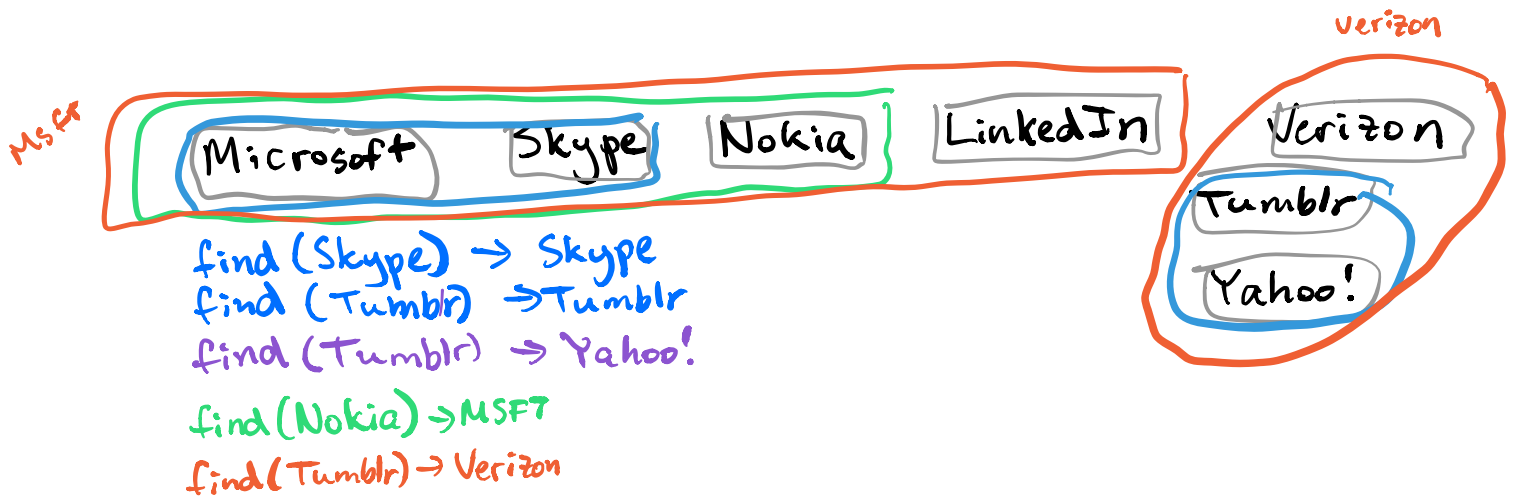


Disjoint Set

Definition: no item is in more than one set
Collection of disjoint sets is a partition

Operations:

1. Union merges 2 sets into 1.
2. Find given an item, find which set it is in



EECS 281, Week 3: Wednesday

FibonacciIterator

Implement `FibonacciIterator`, a class that represents an input iterator that generates Fibonacci numbers. Specifically, implement at least this public API to your class:

- `FibonacciIterator()`, a default constructor that will start by generating the first fibonacci number.
- `FibonacciIterator(size_t n)`, a non-default constructor that will automatically advance to n th fibonacci number.
- Operator `*`, which returns the next Fibonacci number.
- Operator `++`, which advances the iterator to the next Fibonacci number.
- Operator `!=`, which compares two iterators.



Continued on the next page...

EECS 281, Week 3: Wednesday

Implement the `main` of a program that uses `FibonacciIterator` to fill a vector with 100 first Fibonacci numbers and then prints the contents of the vector to `stdout`.

```
int main() {
```

When finished, take a picture of your implementation and email it to Maxim