

Today's announcements:

MP6 available, due , 11:59p.

Hashing Miscellaneous Discussion –

Which collision resolution strategy is better?

- Big records –

- Structure speed –

What structures do hash tables replace for us?

There is a constraint on Keyspaces for BST that does not affect hashing...

Why do we talk about balanced BST if hashing is so great?

Secret mystery data structure

YES

NO

ADT - _____

insert

remove

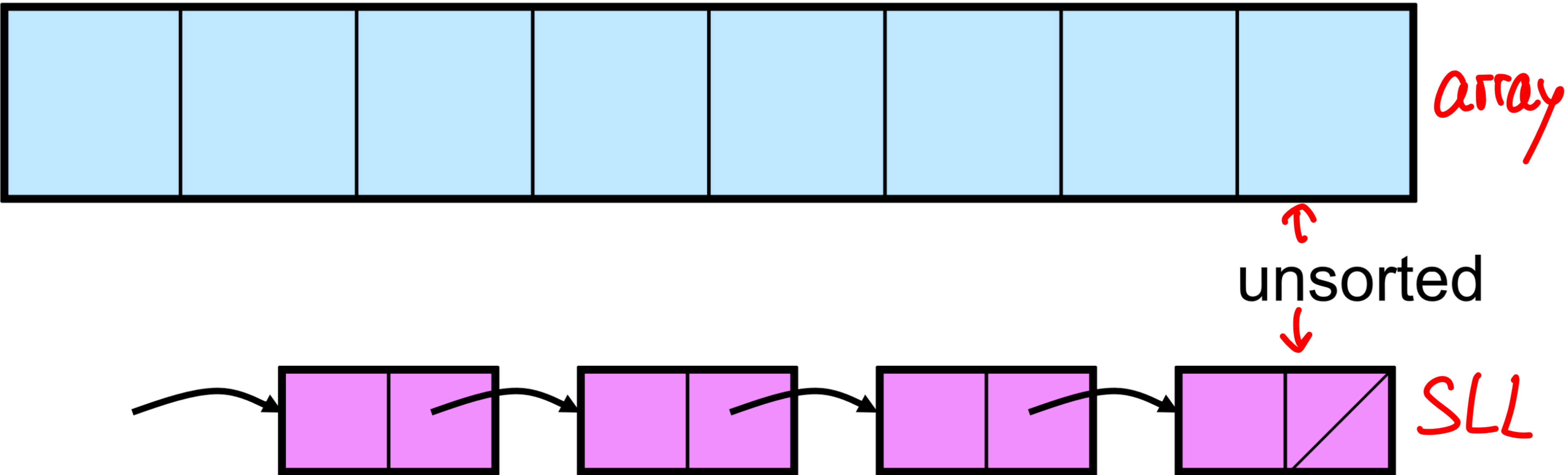
getSize

Priority Queue ADT:

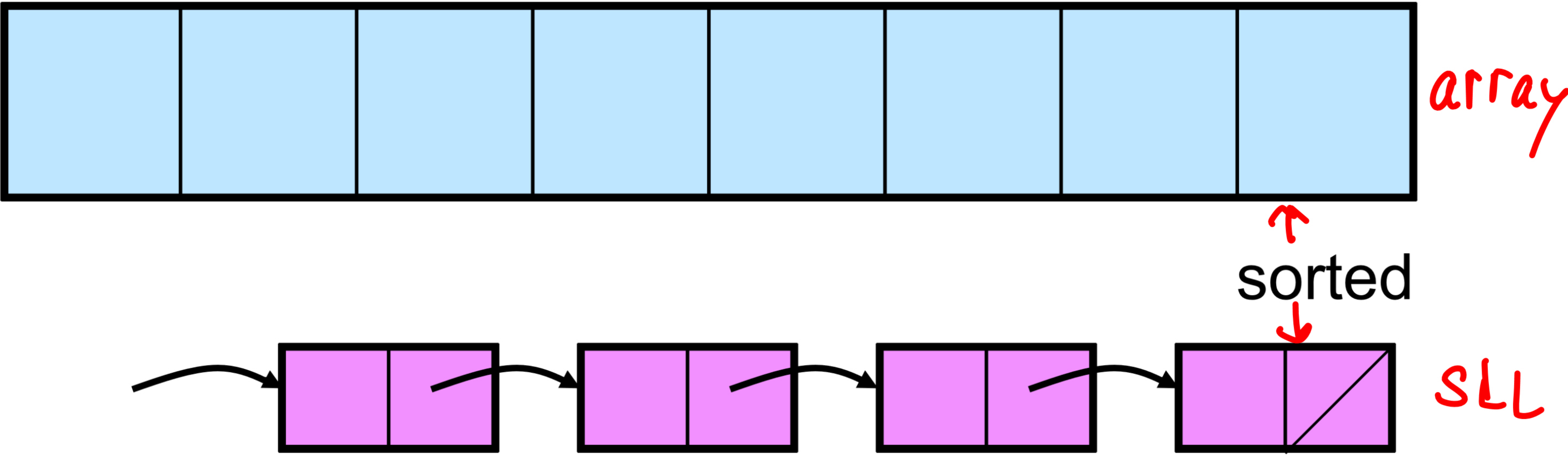
insert removeMin

implementation

$O(n)$	$O(n)$
$O(1)$	$O(n)$

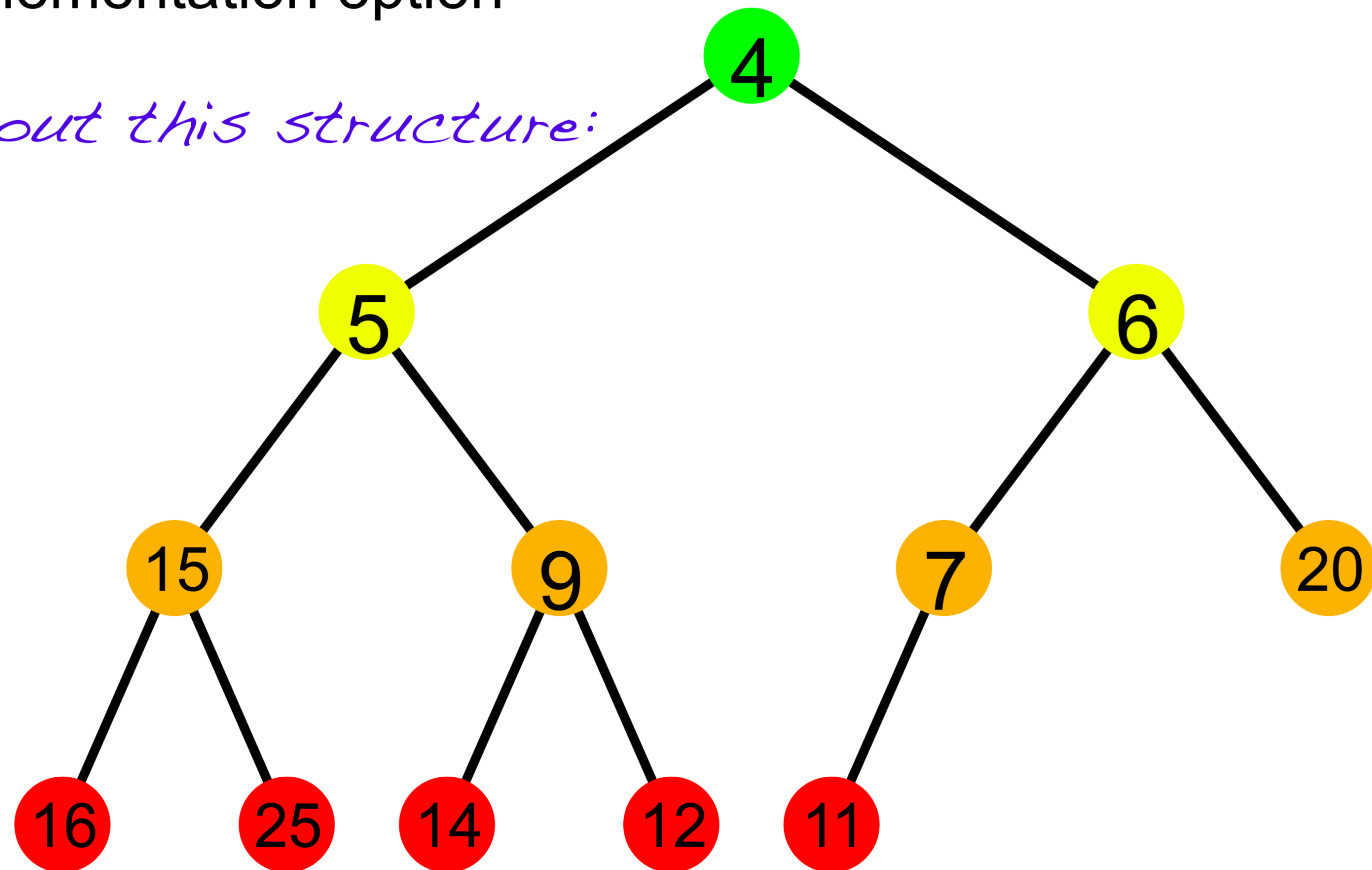


$O(\log n)$	$O(1)$
$O(\log n)$	$O(1)$

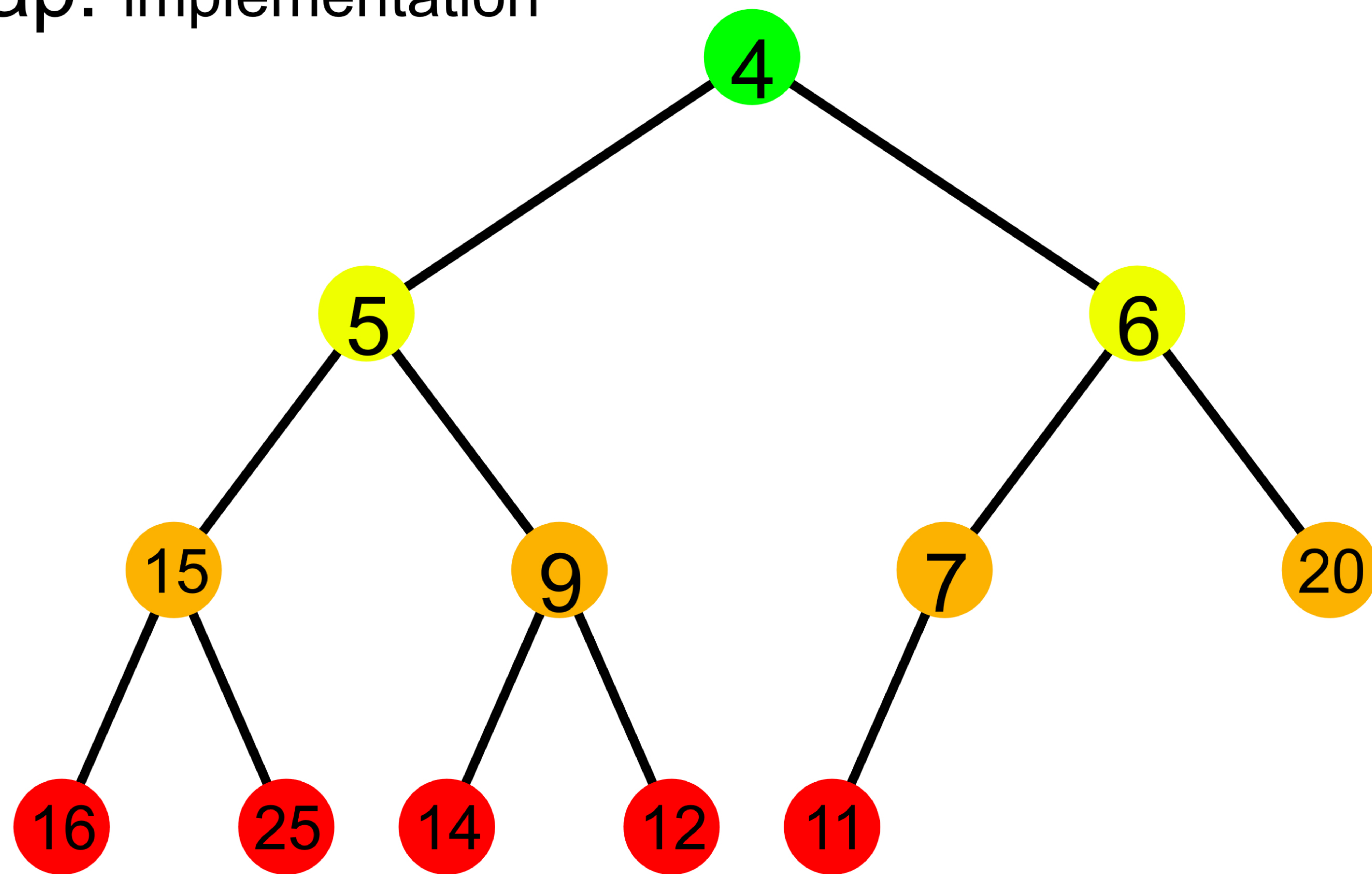


Priority Queue: another implementation option

Tell me everything you can about this structure:

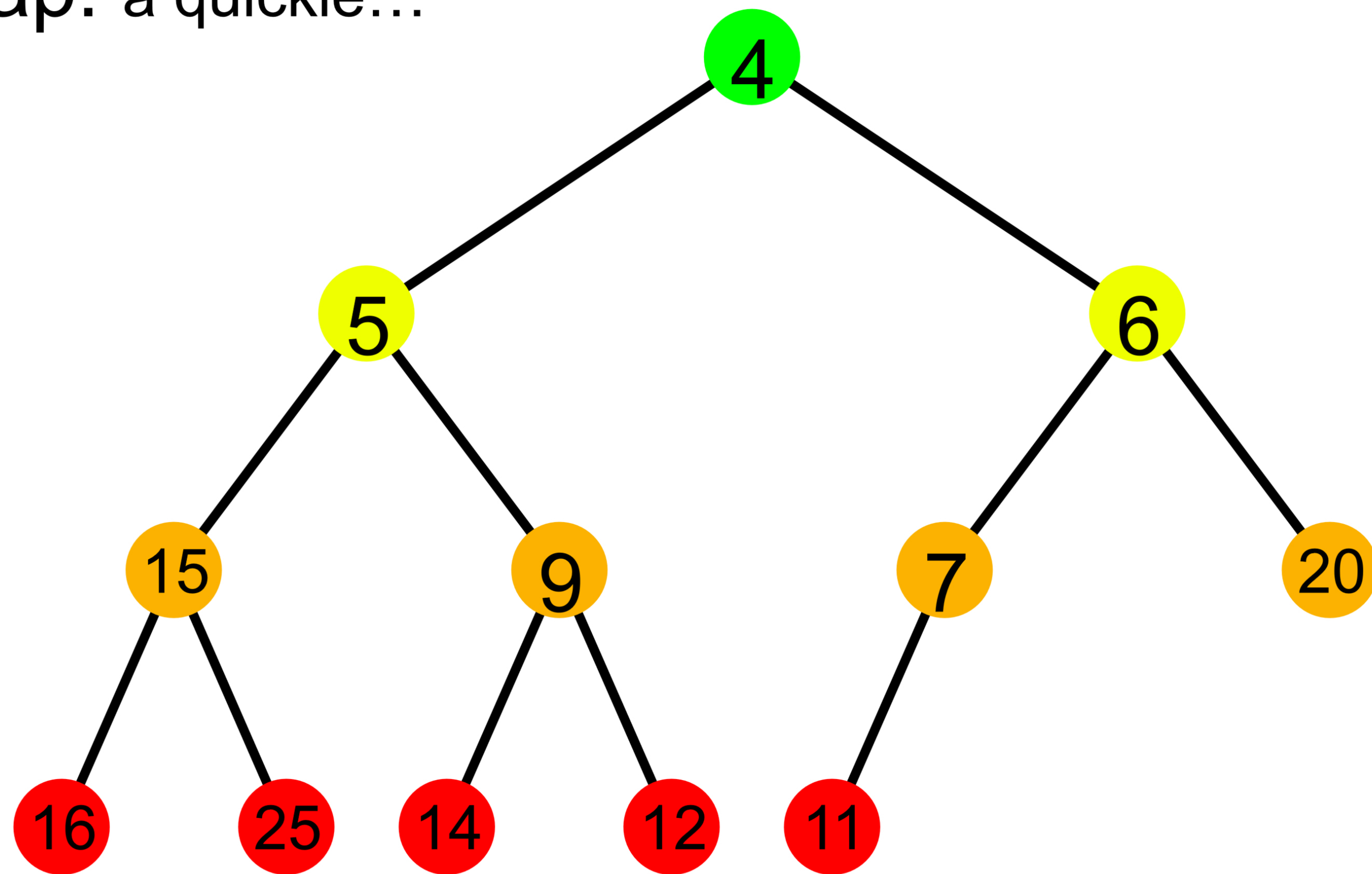


(min)Heap: implementation



4	5	6	15	9	7	20	16	25	14	12	11
---	---	---	----	---	---	----	----	----	----	----	----

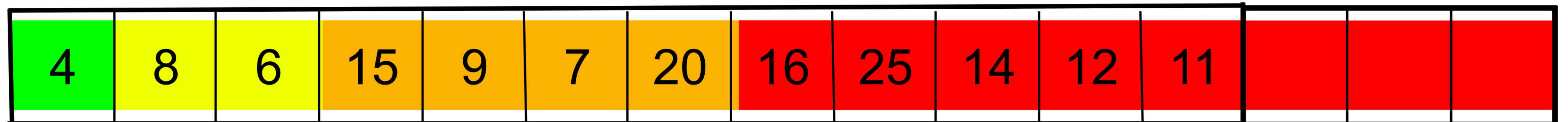
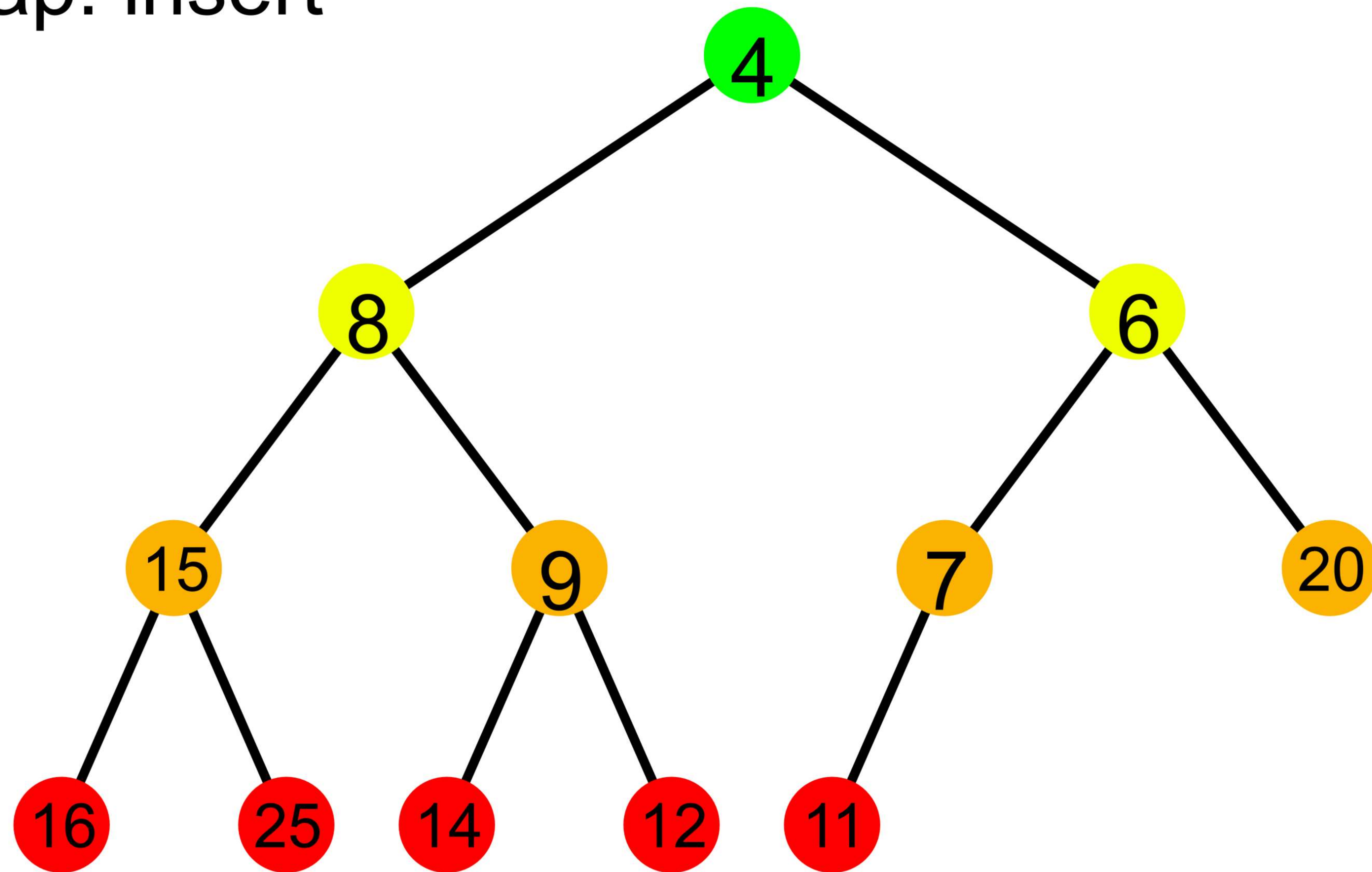
(min)Heap: a quickie...



4	5	6	15	9	7	20	16	25	14	12	11
---	---	---	----	---	---	----	----	----	----	----	----

What is the max height of a complete tree containing n nodes?

(min)Heap: insert



<http://www.cs.usfca.edu/~galles/JavascriptVisual/Heap.html>

<http://people.ksp.sk/~kuko/bak/index.html>

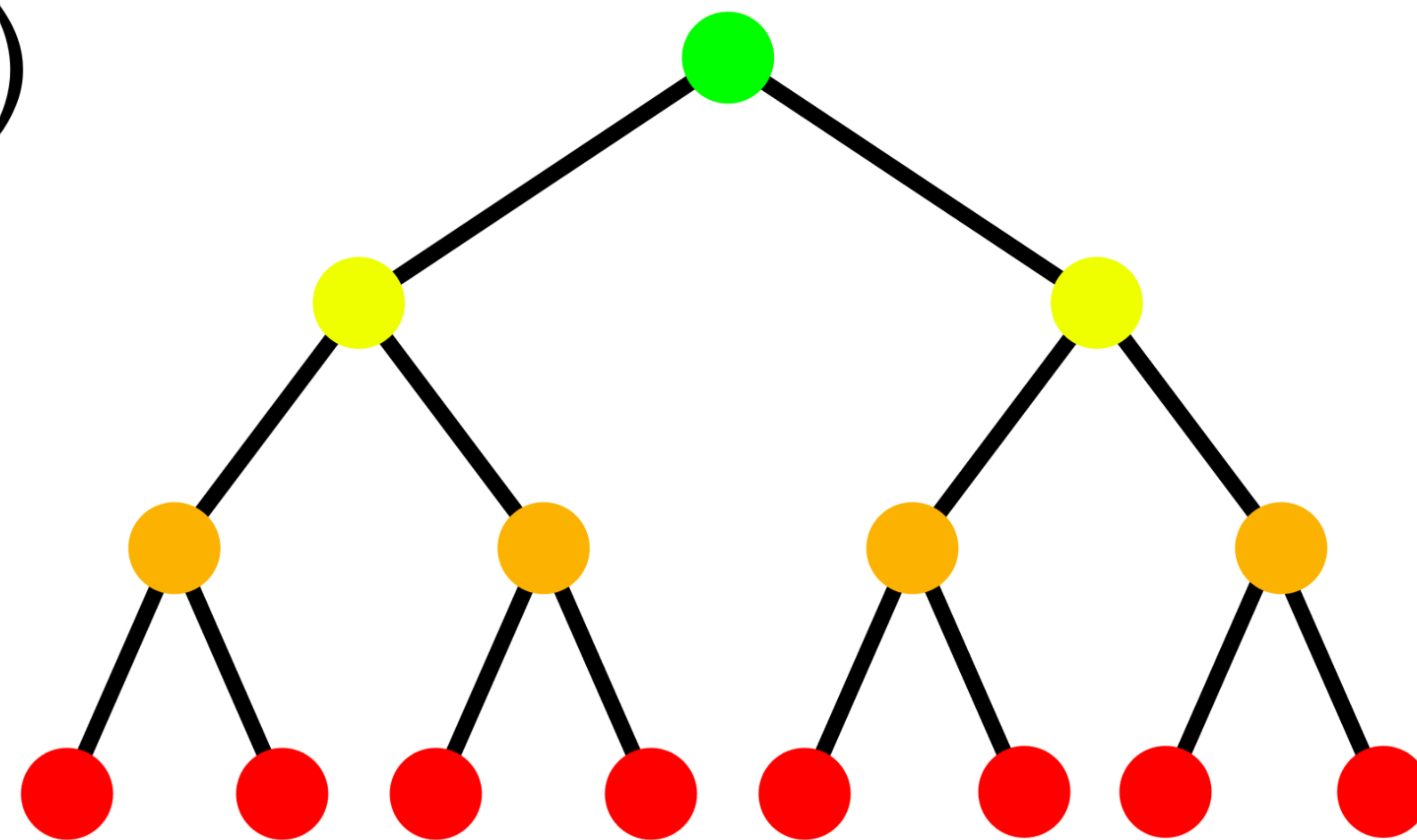
Code:

```
template <class T>
void Heap<T>::insert(const T & key) {

    if (size==capacity) growArray();
    size++;
    items[size] = key;
    heapifyUp(size);

}
```

growArray()



Code:

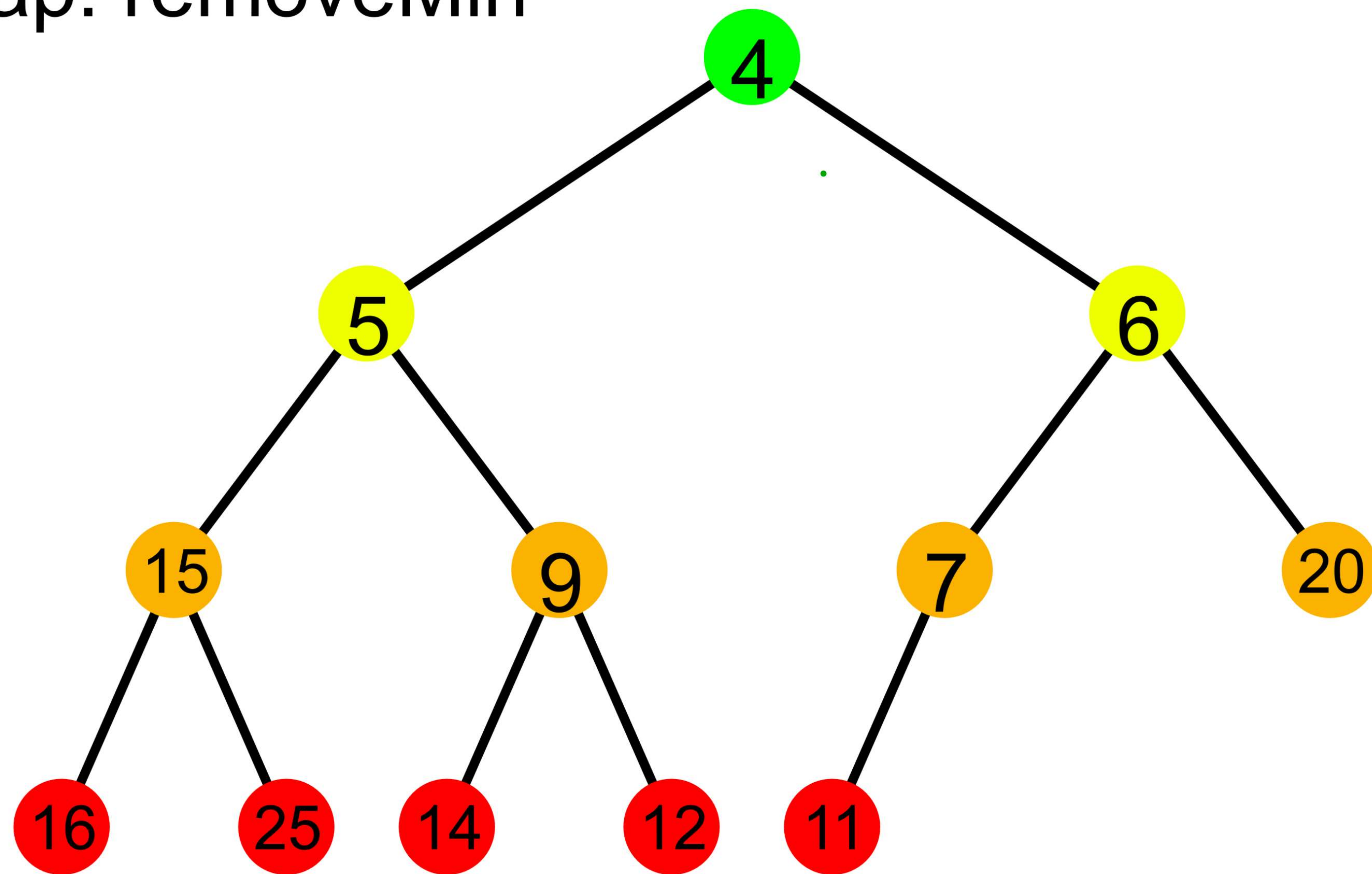
```
template <class T>
void Heap<T>::insert(const T & key) {

    if (size==capacity) growArray();
    size++;
    items[size] = key;
    heapifyUp(size);

}
```

```
template <class T>
void Heap<T>::heapifyUp(int cIndex) {
    if (cIndex > ____){
        if (items[cIndex] ____ items[parent(cIndex)] {
            swap(____, ____);
            heapifyUp(____);
        }
    }
}
```


(min)Heap: removeMin



4	5	6	15	9	7	20	16	25	14	12	11
---	---	---	----	---	---	----	----	----	----	----	----