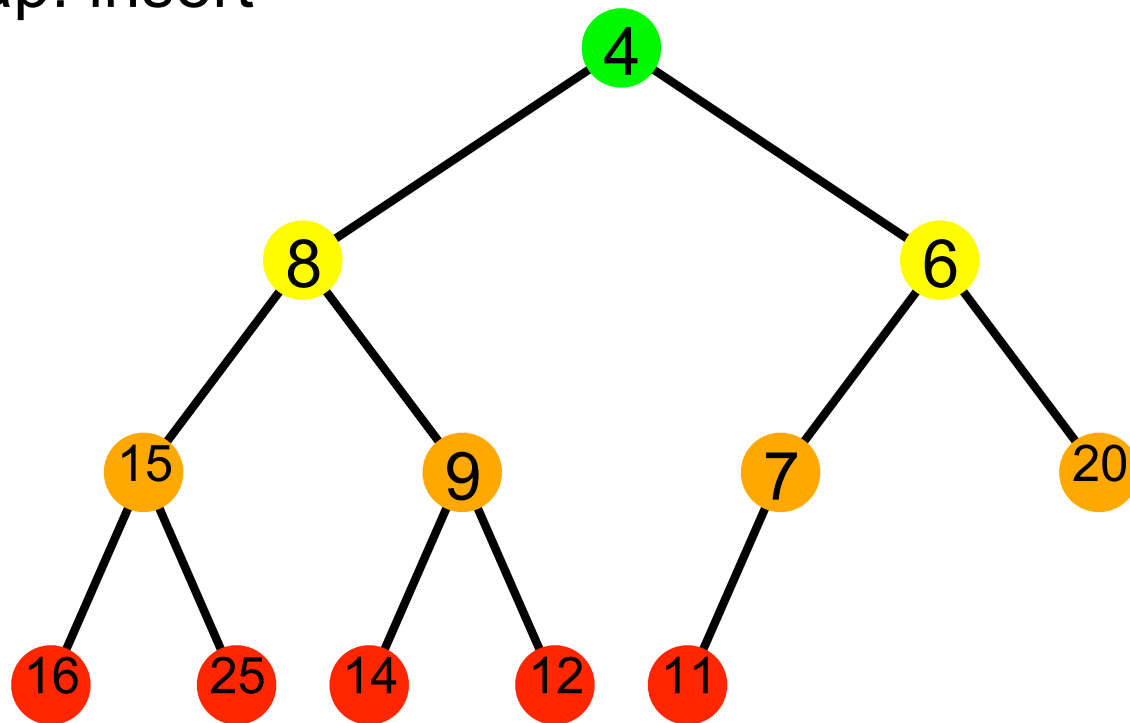


Today's announcements:

MP6 available, due 11/15, 11:59p.

(min)Heap: insert



4	8	6	15	9	7	20	16	25	14	12	11			
---	---	---	----	---	---	----	----	----	----	----	----	--	--	--

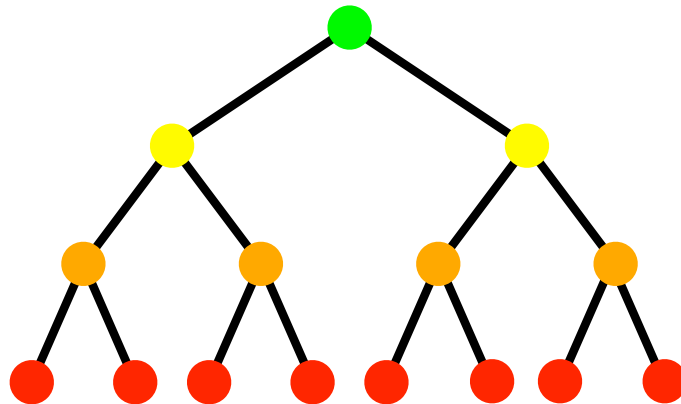
Code:

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

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

}
```

growArray()



Code:

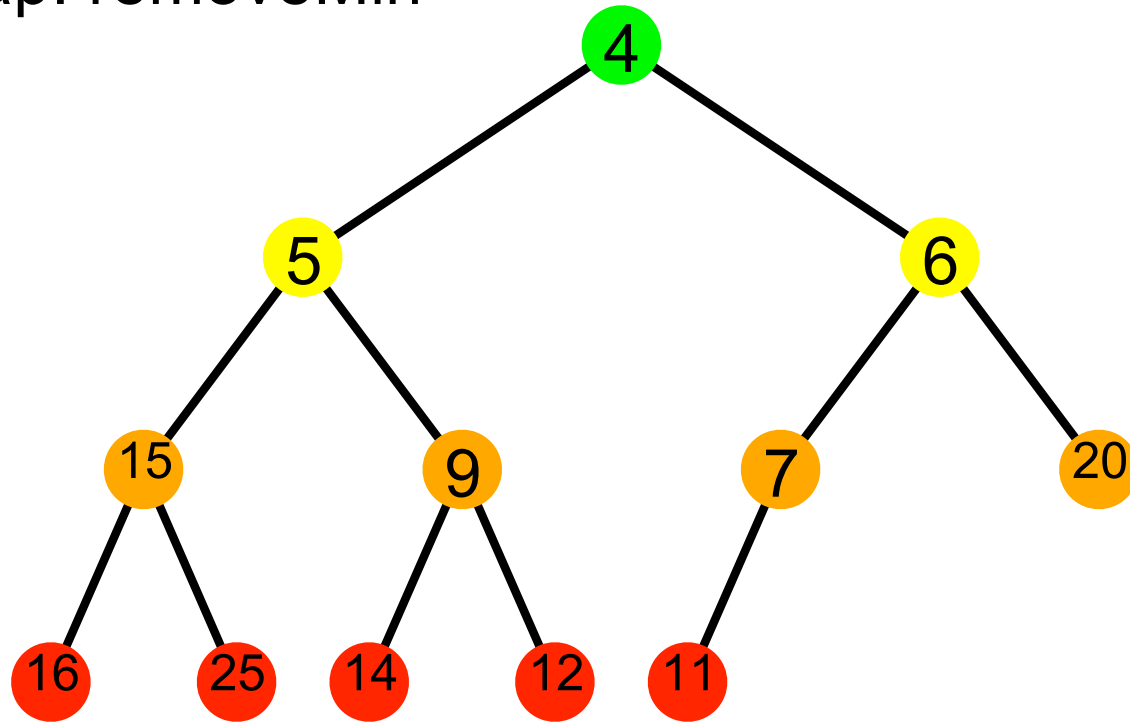
```
template <class T>
void Heap<T>::insert(const T & key){

    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



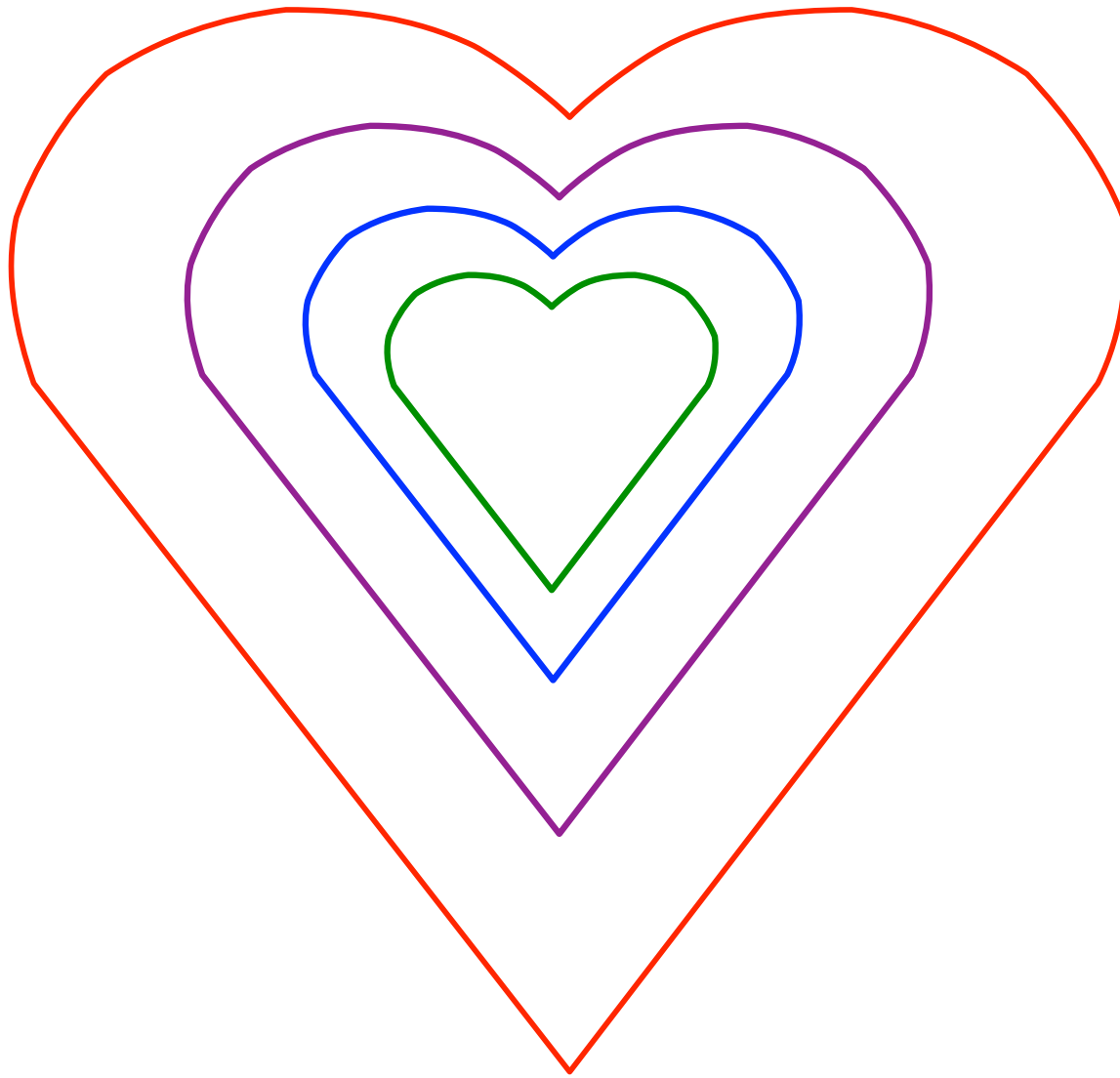
4	5	6	15	9	7	20	16	25	14	12	11
---	---	---	----	---	---	----	----	----	----	----	----

Code:

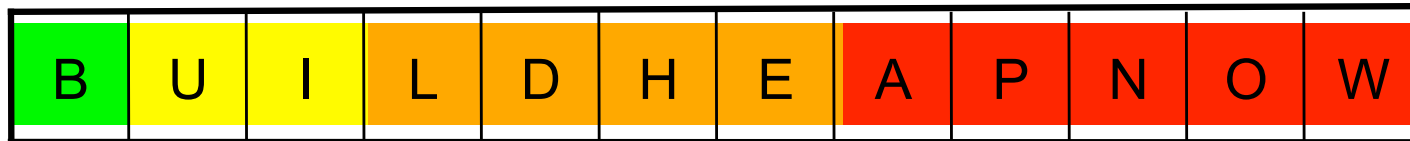
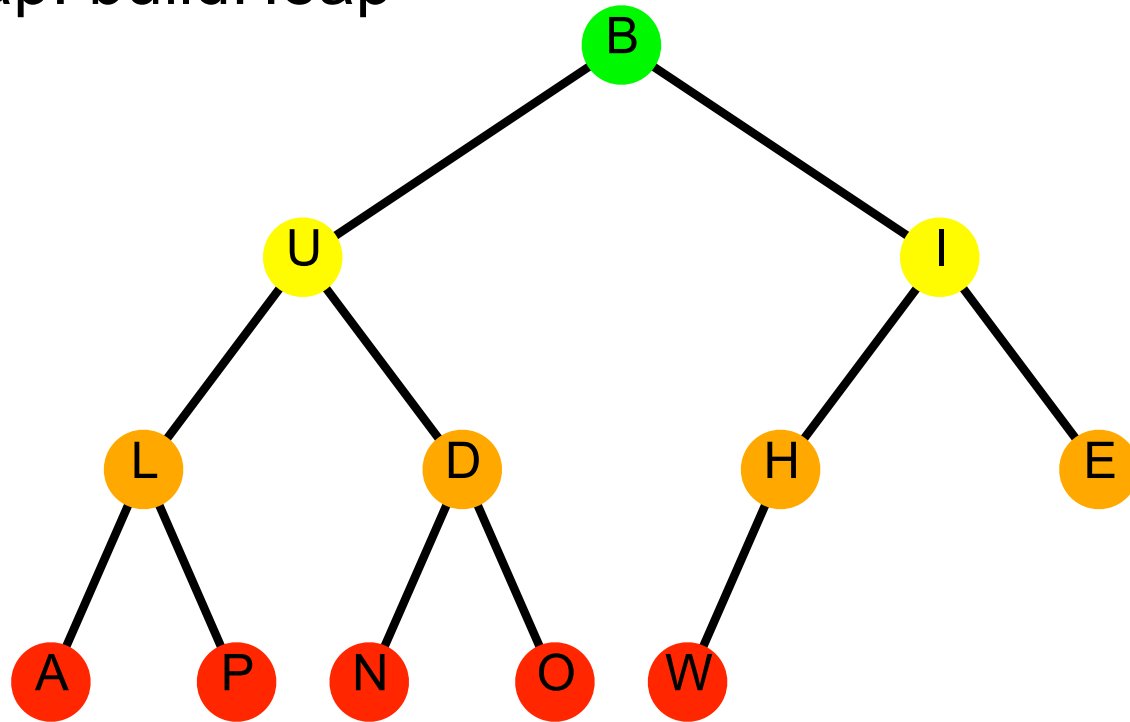
```
template <class T>
T Heap<T>::removeMin() {
    T minVal = items[1];
    items[1] = items[size];
    size--;
    heapifyDown(1);
    return minVal;
}
```

```
template <class T>
void Heap<T>::heapifyDown(int cIndex) {
    if (hasAChild(cIndex)) {
        minChildIndex = minChild(cIndex);
        if (items[cIndex] ____ items[minChildIndex] {
            swap(_____, _____);
            _____;
        }
    }
}
```

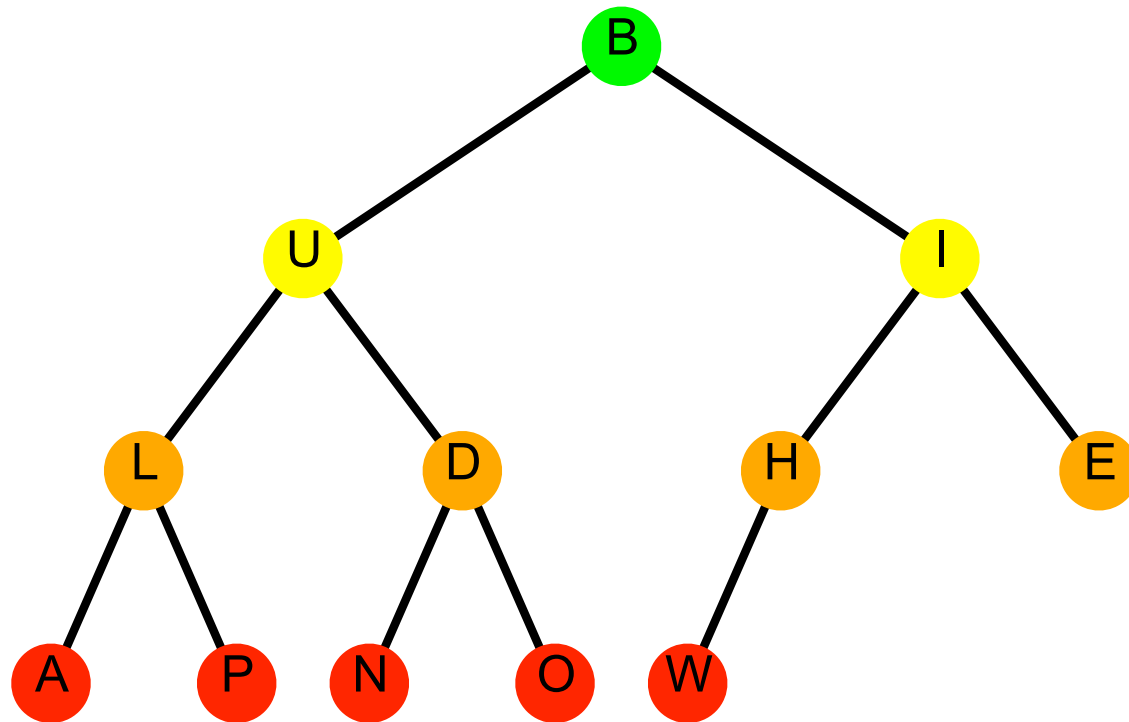
What have we done?



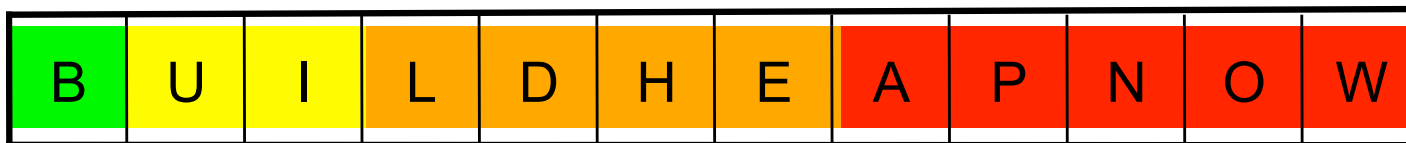
(min)Heap: buildHeap



(min)Heap: buildHeap - 3 alternatives



1. Sort the array:



2.

```
template <class T>
void Heap<T>::buildHeap() {
    for (int i=2; i<=size; i++)
        heapifyUp(i)
}
```

3.

```
template <class T>
void Heap<T>::buildHeap() {
    for (int i=parent(size); i>0; i--)
        heapifyDown(i)
}
```