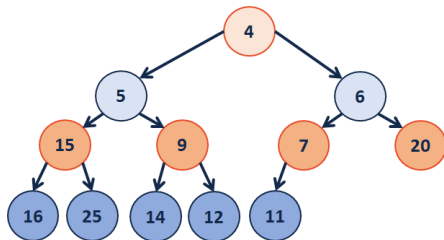


### Implementing a (min)Heap as an Array



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

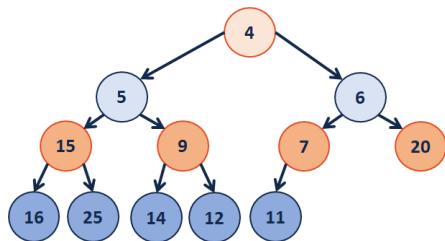
```

1  template <class T>
2  void Heap<T>::_heapifyDown(int index) {
3      if ( !_isLeaf(index) ) {
4          T minChildIndex = _minChild(index);
5          if ( item_[index] > item_[minChildIndex] ) {
6              std::swap( item_[index], item_[minChildIndex] );
7              _heapifyDown( minChildIndex );
8          }
9      }
10 }

```

### Heap Operation: removeMin / heapifyDown insert / heapifyUp

### Heap Operation: removeMin / heapifyDown:



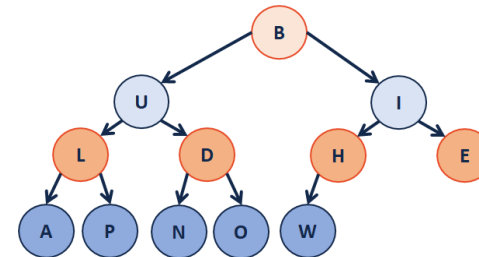
-	4	5	6	15	9	7	20	16	25	14	12	11			
---	---	---	---	----	---	---	----	----	----	----	----	----	--	--	--

```

Heap.hpp (partial)
1  template <class T>
2  void Heap<T>::_removeMin() {
3      // Swap with the last value
4      T minValue = item_[1];
5      item_[1] = item_[size_];
6      size--;
7
8      // Restore the heap property
9      heapifyDown();
10
11     // Return the minimum value
12     return minValue;
13 }

```

### Q: How do we construct a heap given data?



-	B	U	I	L	D	H	E	A	P	N	O	W			
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--	--

### Ideas

- 1)
- 2)
- 3)

**Running Time?**

**Theorem:** The running time of buildHeap on array of size  $n$  is:

\_\_\_\_\_.

**Strategy:**

**Define  $S(h)$ :**

Let  $S(h)$  denote the sum of the heights of all nodes in a complete tree of height  $h$ .

$S(0) =$

$S(1) =$

$S(h) =$

**Proof of  $S(h)$  by Induction:**

**Finally, finding the running time:**

**CS 225 – Things To Be Doing:**

1. mp\_traversals out EC due Monday after break.
2. lab\_huffman is out tomorrow, due Monday after break.
3. final project groups due today!
4. Daily POTDs are ongoing :)