A Heap Data Structure
(specifically a minHeap in this example, as the minimum element is at the root)

Given an index $i$, its parent and children can be reached in $O(1)$ time:
- leftChild := $2i$
- rightChild := $2i + 1$
- parent := floor($i / 2$)

Formally, a complete binary tree $T$ is a minHeap if:

- $T = \{\}$ or

- $T = \{r, T_L, T_R\}$ and $r$ is less than the roots of $T_L, T_R$ and $T_L, T_R$ are minHeaps

Inserting into a Heap

How do we complete this code?

Running time of insert?

Heap Operation: removeMin / heapifyDown:
Heap.hpp (partial)

```cpp
template <class T>
void Heap<T>::_removeMin() {
    // Swap with the last value
    T minValue = item_[1];
    item_[1] = item_[size_-1];
    size_--;
    // Restore the heap property
    heapifyDown(1);
    // Return the minimum value
    return minValue;
}
```

```cpp
template <class T>
void Heap<T>::_heapifyDown(size_t index) {
    if ( !_isLeaf(index) ) {
        size_t minChildIndex = _minChild(index);
        if ( item_[index] > item_[minChildIndex] ) {
            std::swap( item_[index], item_[minChildIndex] );
            _heapifyDown( ________________ );
        }
    }
}
```

Q: How do we construct a heap given data?

![Diagram of a heap](image)

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:

Heap.cpp (partial)

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

Running Time?

CS 225 – Things To Be Doing:

1. Mp_mosaics due date: Today
2. Daily POTDs are ongoing!