#29: Heap Operations
March 29, 2019 · Fagen-Ulmschneider, Zilles

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

![Heap Diagram]

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

![Heap Insertion Diagram]

### Heap Operations: removeMin / heapifyDown:

```cpp
template <class T>
void Heap<T>::_insert(const T & key) {
    // Check to ensure there’s space to insert an element
    // ...if not, grow the array
    if ( size_ == capacity_ ) { _growArray(); }
    // Insert the new element at the end of the array
    item_[++size] = key;
    // Restore the heap property
    _heapifyUp(size);
}
```

```
template <class T>
void Heap<T>::_heapifyUp(int index) {
    if ( index > ___ ) {
        if ( item_[index] < item_[ parent(index) ] ) {
            std::swap( item_[index], item_[ parent(index) ] );
            _heapifyUp( ___ );
        }
    }
}
```

How do we complete this code?

Running time of insert?
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 $h$.

$S(0) = \quad S(1) = \quad S(h) = \quad$

Proof of $S(h)$ by Induction:

Finally, finding the running time:

CS 225 – Things To Be Doing:

1. Theory Exam 3 starts on Thursday (practice exam online!)
2. MP5 due date: Monday, April 1st
3. lab_hash is due Sunday, March 31st
4. Daily POTDs are ongoing!