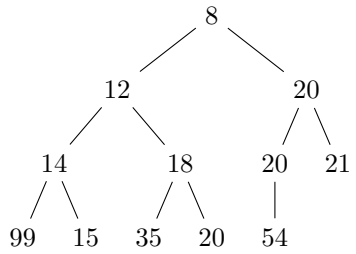
Heap

- Binary Search Tree
- All but the bottom level is complete.
- From every node x, x is less than it's two children.
- Member functions.
  - top(T) = *8*
  - insert(T, x)
  - remove(T) = *T = the top of the heap*
- Maintenance Functions
  - Percolate Up, during insertion.
    - Let the item bubble up.
  - Percolate Down, during removal.
    - Like a stone sinking through a viscous liquid.



D.S. ArrayHeap

```
class ArrayHeap {
    T *data;
    int m_max, m_size;

public:
    const T& top() {
        if (m_size != 0) {
            return m_data[0];
        } else {
            cerr << "Shit."
        }
    }

    void insert(const T& x) {
```

```
        if (m_max == m_size) {
            grow();
        }

        int hole = m_size;
        m_size++;

        while (hole > 0 && x < m_data[(hole- 1) / 2]) {
            m_data[hole] = m_data[(hole - 1) / 2];
            hole = (hole - 1)/2;
        }

        m_data[hole] = x;
    }

    void remove() {
        if (m_size == 0) { return; }
        int hole = 0;
        m_size--;

        We can continue this over break.
    }
};
```