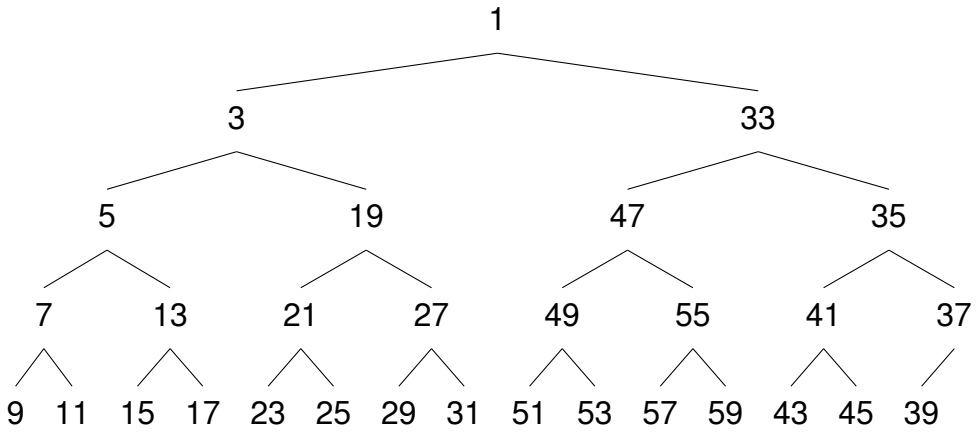
DSA Homework #5

5.1 Heap and Hash

1.



2.

```
class stack{
  Int counter = 0;
  priority_queue<tuple<Int,Int>>pQueue;
  function push(var){
    pQueue.insert( counter, var);
    counter -= 1;
  }
  function pop(){
    pQueue.removeMin();
  }
  function top(){
    return pQueue.min().value;
  }
  function size(){
    return pQueue.size();
  }
  function empty(){
    return pQueue.empty();
  }
}
```

3.

```
function findLeq( heapRoot, k){
  List leqList;
  if( heapRoot -> value <= k ){
    leqList -> insert( k );
    for( child in heapRoot -> childs ){
```

```
            leqList -> insert( findLeq( chind, k ) );
          }
        }
        return leqList;
      }
```

4.

5.

```
      function findDiffCharPosition( str1, str2 ){
        int strLength = str1.length();
        Int upper_bound = strLength - 1;
        Int lower_bound = 0;
        while( upper_bound != lower_bound ){
          Int mid = ( upper_bound + lower_bound ) / 2;
          if( postfixHash( str1, strLength - mid ) ==
              postfixHash( str2, strLength - mid ) ){
            upper_bound = mid;
          }else{
            lower_bound = mid + 1;
          }
        }
        return lower_bound;
      }
```