Using Induction to Show Correctness

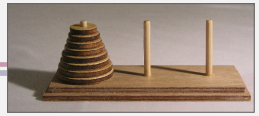
- use induction with a *loop invariant* to show correctness of a loop
  - a loop invariant is a boolean statement about the solution so far
    - true at the beginning of each loop iteration
    - when the loop exits, used to show that the final result is correct
- use induction to show correctness of recursion

Insertion Sort

```
public static void sort(int[] arr) {
  for ( int i = 1 ; i < arr.length ; i++) {
    // arr[0..i-1] (inclusive) is sorted in increasing order
    int elt = arr[i]; // current element to put in place

    // shift - move elements of arr[0..i-1] that are
    // greater than elt one spot to the right
    int shift = i - 1;
    for ( ; shift >= 0 && arr[shift] > elt ;
          shift = shift - 1) {
      arr[shift + 1] = arr[shift];
    }

    // put element in place
    arr[shift + 1] = elt;
  }
}
```

- let P(i) be the statement that `arr[0..i-1]` (inclusive) is sorted in increasing order

Towers of Hanoi

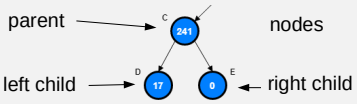


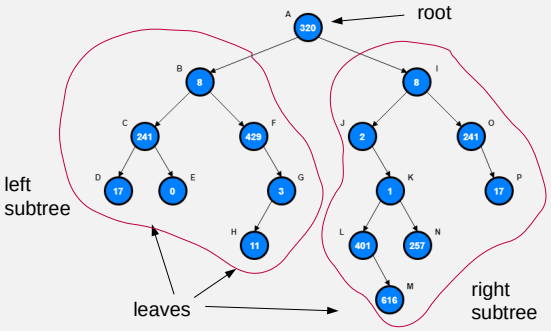
```
public static void hanoi(int n,
                         int from, int to, int spare) {
  // if there is only one disk, simply move it
  // otherwise
  // move the top n-1 disks out of the way (onto spare) so that
  // the nth disk can be moved
  // move the nth disk
  // move the n-1 disks from the spare to the final goal

  if (n == 1) {
    System.out.println("move disk from " + from + " to " + to);
  } else {
    hanoi(n - 1, from, spare, to);
    System.out.println("move disk from " + from + " to " + to);
    hanoi(n - 1, spare, to, from);
  }
}
```

- let P(n) be the statement that this code gives a valid solution for the towers of hanoi with n disks, $n \geq 1$

https://commons.wikimedia.org/wiki/File:Tower_of_Hanoi.jpeg

Binary Trees





https://emre.me/data-structures/binary-tree/

Binary Trees

sum of the elements in the tree –

```
sum(leaf) = leaf.elt
sum(node) = sum(left(node))+sum(right(node))+node.elt
```