COP 3502 9/8/2025

~~0)~~ P1 Sol, Q1 Sol

~~1)~~ Towers of Hanoi

2) Binary Search

3) Brute Force

To solve Towers(n)

1. Solve Towers(n-1)
2. Move ~~nth~~ bottom disk
3. Solve Towers(n-1)

Often times, recursive functions need extra parameters to do their task than corresponding iterative functions

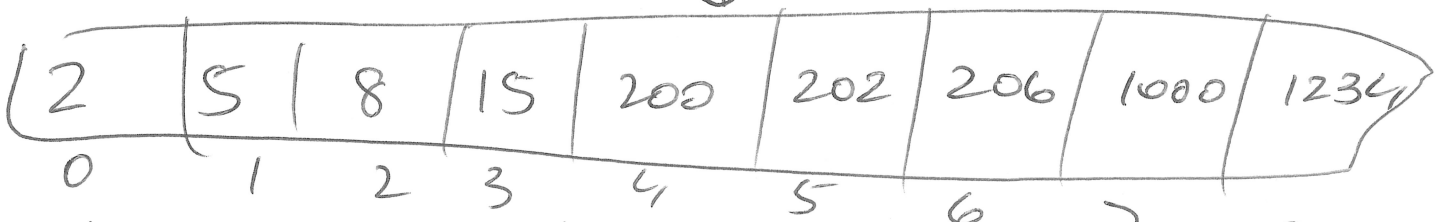
```
Towers(int n, int start, int end) {
    if (n > 0) {                          // not start ^ not end
        Towers(n-1, start, other);
        printf("Moving disk %d from tower %d to tower %d\n", n, start, end);
        Towers(n-1, other, end);
    }
}
```

FYI num of moves $= 2^n - 1$ (we'll prove when we look at recurrence relations.)

Binary Search

Problem: Given a sorted list of items and an item to find, find it or report it's not in the list

17



to search in whole array binsearch(array, 0, n-1, target); &

```
int binsearch(int * array, int low, int high, int target) {
    // not search space
    if (low > high) return -1;
    int mid = (low + high)/2;
    if (target < array[mid])
        return binsearch(array, low, mid-1, target);
    else if (target > array[mid])
        return binsearch(array, mid+1, high, target);
    else
        return mid;
}
```