CSE 3010 – Data Structures & Algorithms

Lecture #11

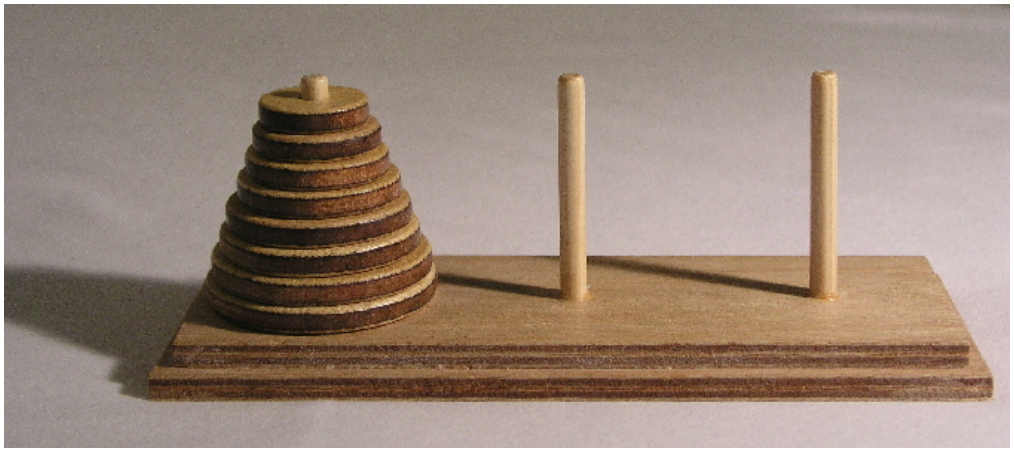
What will be covered today

- Understanding recursion
- Representation of a linked list

n^{th} term of a Fibonacci series

```
int fibonacci(int n) {
    printf("Value of n is %d\n",n);
    if (n <= 1)
        return n;
    else
        return (fibonacci(n - 2) + fibonacci(n - 1));
}
```

Towers of Hanoi



Moving 3 disks take 7 moves
Moving 4 disks take 15 moves
Moving 5 disks take 31 moves
….
Moving n disks take $2^n - 1$ moves

For n = 64, it takes 18,446,744,073,709,551,615 moves
Approximately equal to 584 billion years☺

Towers of Hanoi

```
void tower_hanoi(int n, char source, char dest, char spare) {


        if (n == 1) {
                printf("\nMove disk 1 from pole %c to pole %c",
source, dest);
                return;
    }


        tower_hanoi(n-1, source, spare, dest);


        printf("\nMove disk %d from pole %c to pole %c", n, source,
dest);


        tower_hanoi(n-1, spare, dest, source);
}
```

Rules of recursion

- Must have a base case
- Must change its state and move towards the base case
- Must call itself

Exercises:

1. Reverse a string using recursion (only for practice)
2. Read what `atoi()` function in C does. Write your version of `atoi()` function using recursion

Visit https://runestone.academy/runestone/books/published/pythonds/Recursion/DynamicProgramming.html to get an understanding of different problems that use recursion

Representation of linked list