

---

## Unit 4: Exploring Programming Language Concepts

---

### Unit 4 Seminar: Programming Language Concepts

#### Recursion

##### Task:

One of the classic programming problems that is often solved by recursion is the towers of Hanoi problem. A good explanation and walkthrough are provided by Cormen & Balkcom (n.d.) - the link is in the reading list. (the code they used for their visual example is provided on their website as well).

- Read the explanation, study the code and then create your own version using Python (if you want to make it more interesting you can use asterisks to represent the disks). Create a version that asks for the number of disks and then executes the moves, and then finally displays the number of moves executed.
- What is the (theoretical) maximum number of disks that your program can move without generating an error?
- What limits the number of iterations? What is the implication for application and system security?

##### Answer:

###### 1. Theoretical Maximum Number of Disks:

The maximum number of disks depends on the recursion depth and the stack size allowed by Python. By default, Python sets a recursion limit of 1,000, but this can be adjusted using `sys.setrecursionlimit()`. However, increasing it too much may cause memory errors.

###### 2. Limits of Iterations:

The primary limiting factor is memory, particularly the stack used for recursion. For very high values, the stack may overflow, or memory may run out. Additionally, larger computations will take more time.

### **3. Implications for System Security:**

If input is not validated, extremely high recursion could cause a Denial of Service (DoS) attack by exhausting system resources.

Limiting recursion depth, handling exceptions, and validating inputs ensure stability and security.