Assignment2 – Introduction and Prolog

Problem 2.1 (Towers of Hanoi)

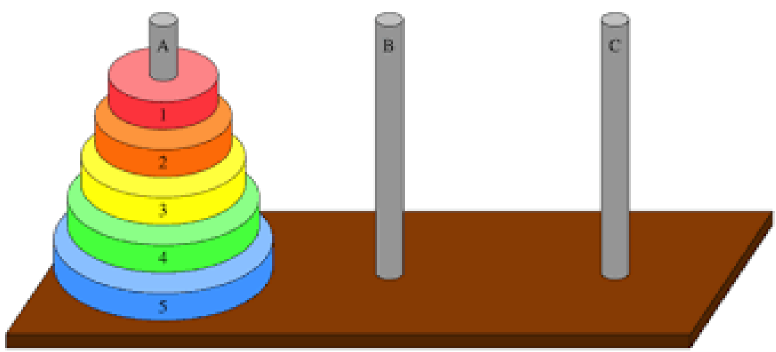
The Towers of Hanoi is a puzzle. It consists of three pegs (A, B, and C) and a number of disks of different sizes, which can slide onto any peg. The puzzle starts with the disks in a stack in ascending order of size on one peg, the smallest at the top, thus making a conical shape. The objective of the puzzle is to move all disks from peg A to peg B, while obeying the following rules:

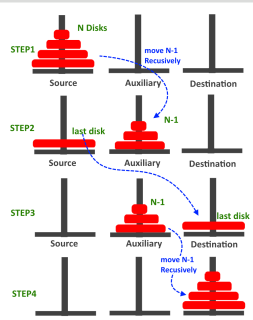
1. only one disk can be moved at a time,

2. each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty peg,

3. no larger disk may be placed on top of a smaller disk.

The idea of the algorithm (for $N > 1$) is to move the top $N-1$ disks onto the auxiliary peg, then move the bottom disk to the destination peg, and finally moving the remaining $N-1$ disks from the auxiliary peg to the destination peg.





1. Write a *Prolog* predicate that prints out a solution for the Towers of Hanoi puzzle. Use the `write(X)` predicate that prints the value of X (X can be simple text or any type of argument) to the screen and `nl` that prints a new line

to write a rule `move(N, A, B, C)` that prints out the solution for moving N disks from peg A to peg B, using C as the auxiliary peg.
Each step of the solution should be of the form "Move top disk from X to Y".
Examples:

```
?- write(hello), write('␣world!'), nl.
hello world!
true.
```

```
?- move(3, left, center, right).
Move top disk from left to center
Move top disk from left to right
Move top disk from center to right
Move top disk from left to center
Move top disk from right to left
Move top disk from right to center
Move top disk from left to center
true ;
false.
```

2. Determine the complexity class of your algorithm in terms of the number of disks N and explain how you computed it.

Problem 2.2 (Mathematical Notation)

Let $\mathbb{N}$ be the set of natural numbers. A *monoid* is a *mathematical* structure $\langle G, \circ, e \rangle$ where G is a *set*, $\circ$ is an *associative* binary *function* on G, and e is the *neutral element* of $\circ$.

Express the following concepts in *mathematical* notation:

1. The set containing all natural numbers
2. The set containing the set of natural numbers
3. The set containing all square numbers
4. The set containing all even natural numbers
5. The set containing all even square numbers
6. The 3-tuple of 0, 1, and 2
7. The n-tuple of all numbers from 0 to $n-1$
8. The set of pairs of natural numbers and their squares
9. The pair of sets of natural numbers and square numbers
10. The monoid of natural numbers under addition
11. The pair of monoids of the natural numbers under addition and under multiplication
12. The set of the monoids of the natural numbers under addition and under multiplication
13. Given a *monoid* $\langle G, \circ, e \rangle$, the set of elements that are not the neutral element
14. Given a *monoid* $\langle G, \circ, e \rangle$, the monoid in which the operation is the same but with left and right argument switched.

Problem 2.3 (Prolog Grammar)

Consider the following partial grammar for a simplified version of Prolog with start symbol P and productions

P	$::=$	C^*	programs: lists of clauses
C	$::=$		clauses: head literal and list of body literals
L	$::=$		literals: predicate symbol applied to list of terms
	$\mid$		or term equality
T	$::=$		terms: function symbol applied to list of terms
I	$::=$	alphanumeric string	identifiers

1. Complete the grammar with appropriate productions.
2. The above grammar uses N^* to indicate a repetition (list) of words derived from N. But often we want a list with a separator, e.g., N, N, N instead of NNN.
   Describe how the production $A ::= B\,N^*\,C$ can be revised to produce a comma-separated list (assuming ‘,‘ is among the terminal symbols).