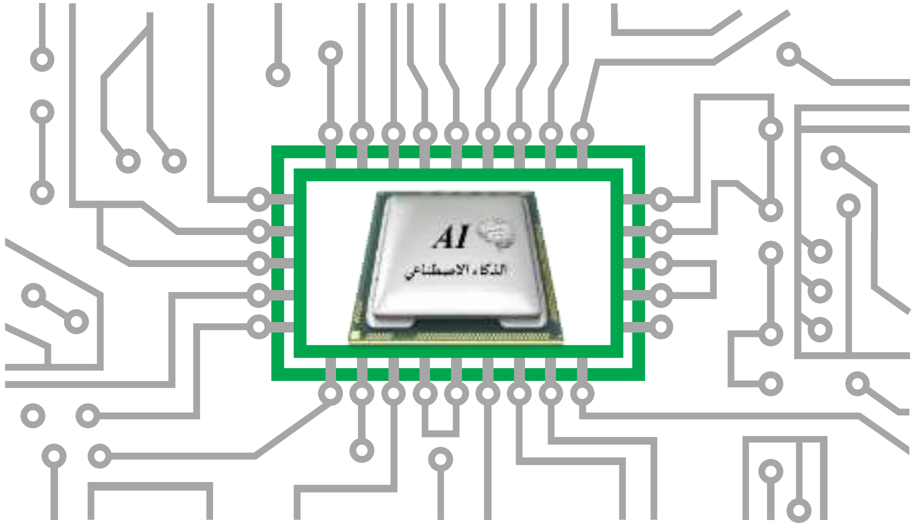
# COE 292
# Introduction to Artificial Intelligence




## Goal Trees and Problem Solving

# Outline

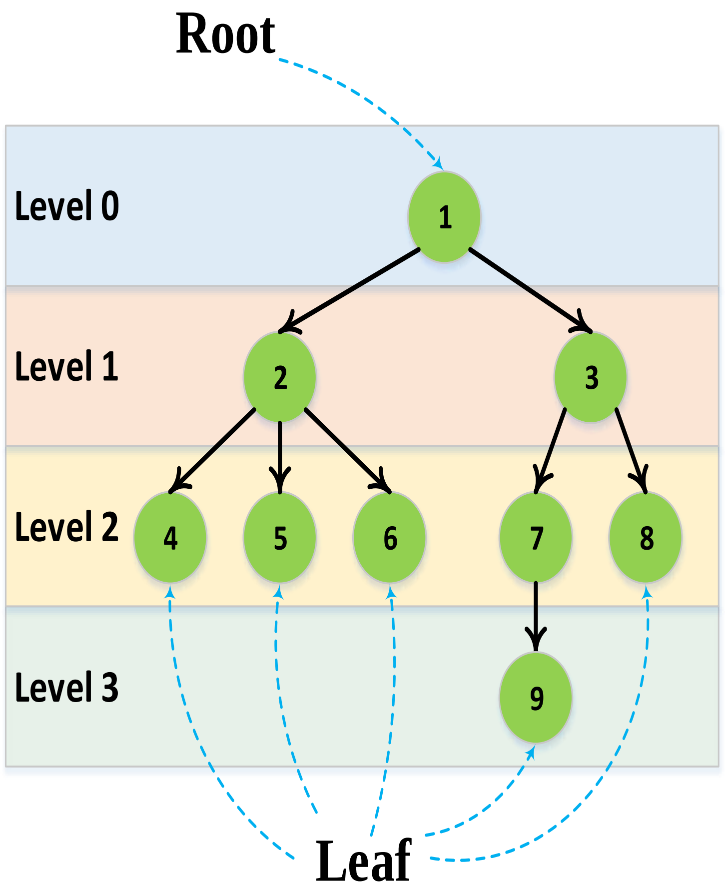
- Problem solving
  - Review
  - Goal trees
- Problem Reduction
  - Tower of Hanoi
  - Symbolic Integration [The start of AI]
- Understanding Goal Trees
- Conclusion

# Problem Solving - Review

- As we concluded from previous lectures, successful problem solving is aided by:
  - Choosing the best representation for the problem, e.g., using visual perception
  - Identifying the initial and goal states
  - Identifying all possible states
  - Exposing constraints

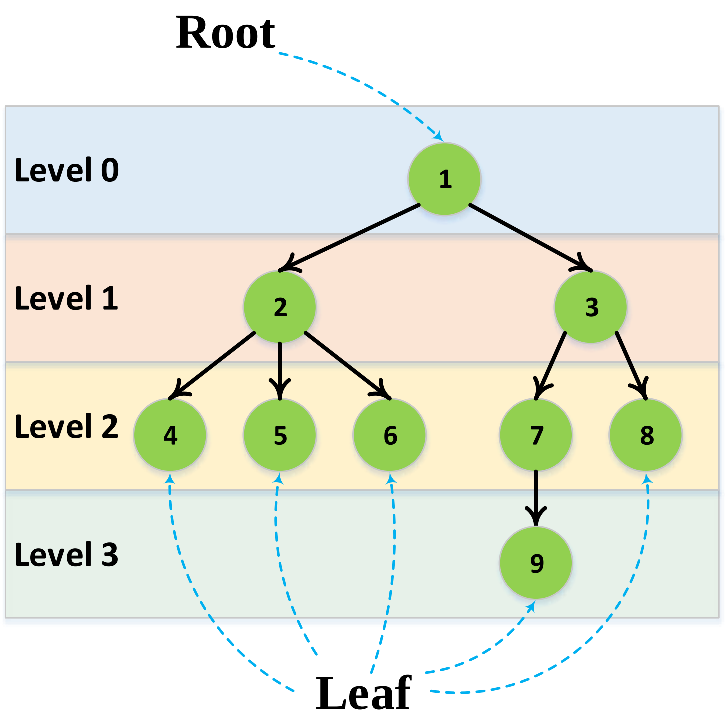
# Problem Solving – Goal Trees

- A *tree* is commonly encountered data shape that allows representing nonlinear hierarchical relationships
- A tree is a collection of structures called nodes connected using edges where a single node represents a value, a *state* or something meaningful



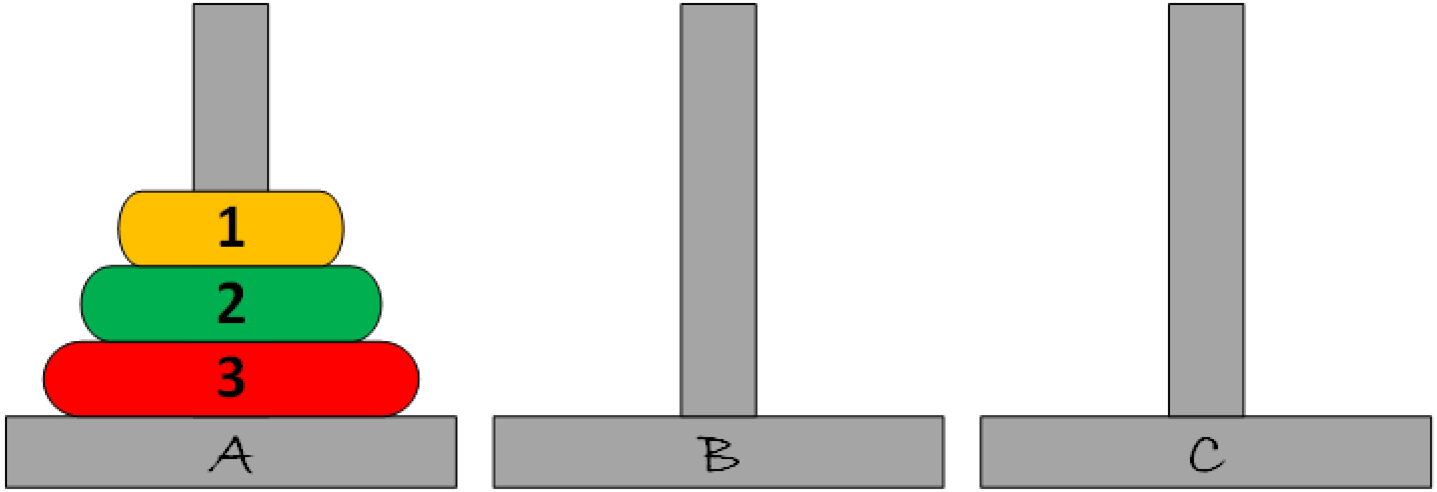
# Problem Solving – Goal Trees

- Properties of trees:
  - A node may have multiple successors
  - Nodes with at least one successor is called a *parent* and its successors are called *children*
  - A node without a child is called a *leaf* node
  - A node that has no parent is called the *root*
  - *Depth* of a tree is the number of levels between the root and the farthest leaf node
    - Depth of shown tree is **3**



# Tower of Hanoi

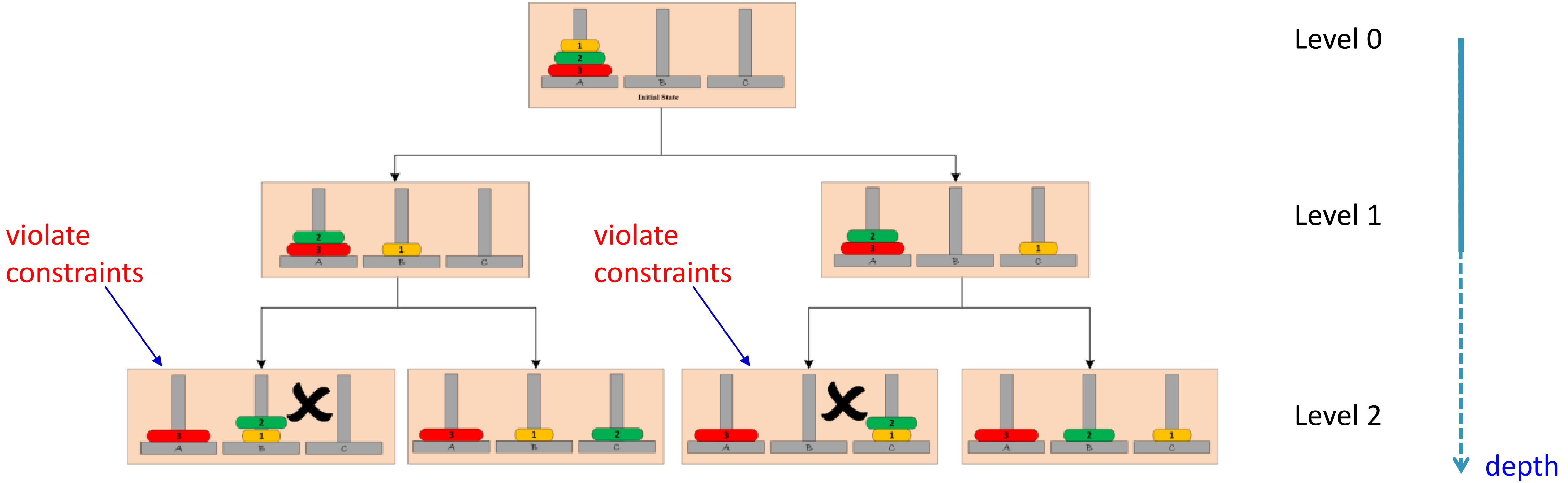
- The objective of the game is to move all ***n*** disks from pin $\mathcal{A}$ over to pin $\mathcal{C}$ but with the following constraints:
  - You cannot place a larger disk onto a smaller disk
  - You can move one disk at a time
- Example ***n*** = 3




**Initial State**

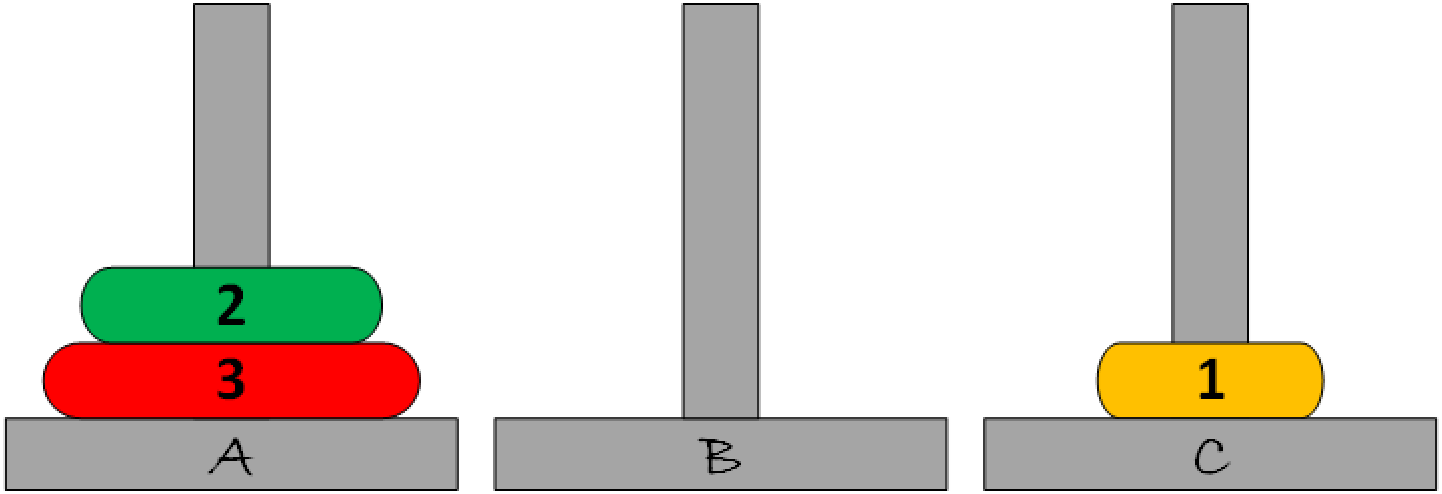
# Example of Tower of Hanoi

- The Tree starts from the Initial state and details the steps on how to reach the goal state



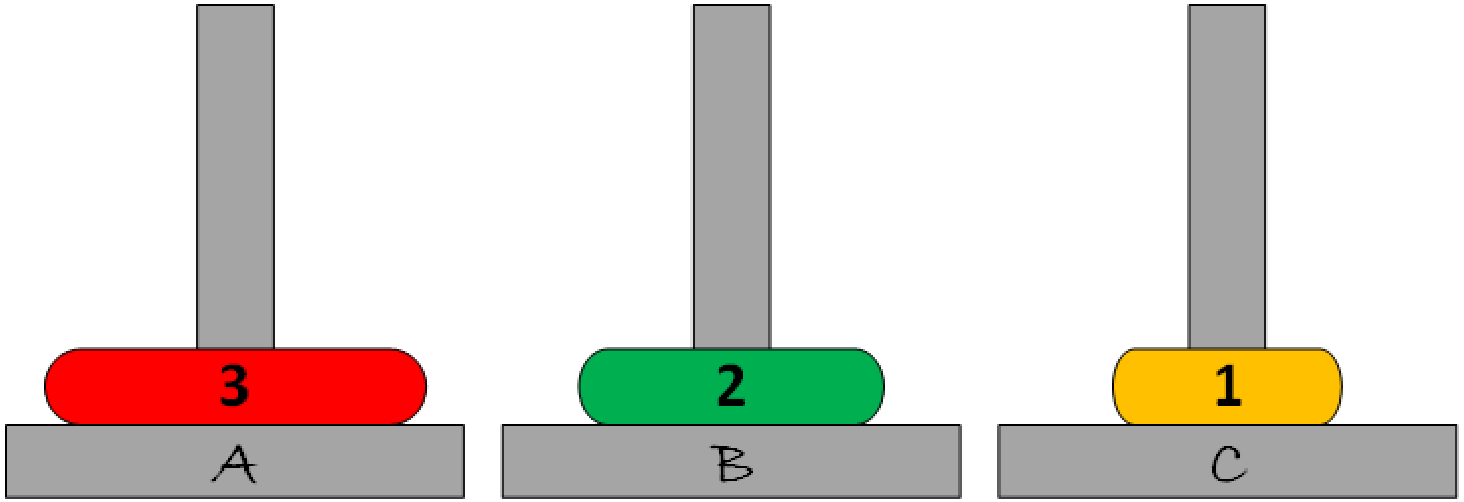

- The above Tree is up to the 2nd level, you can continue until you reach the goal state.

# Example of Tower of Hanoi



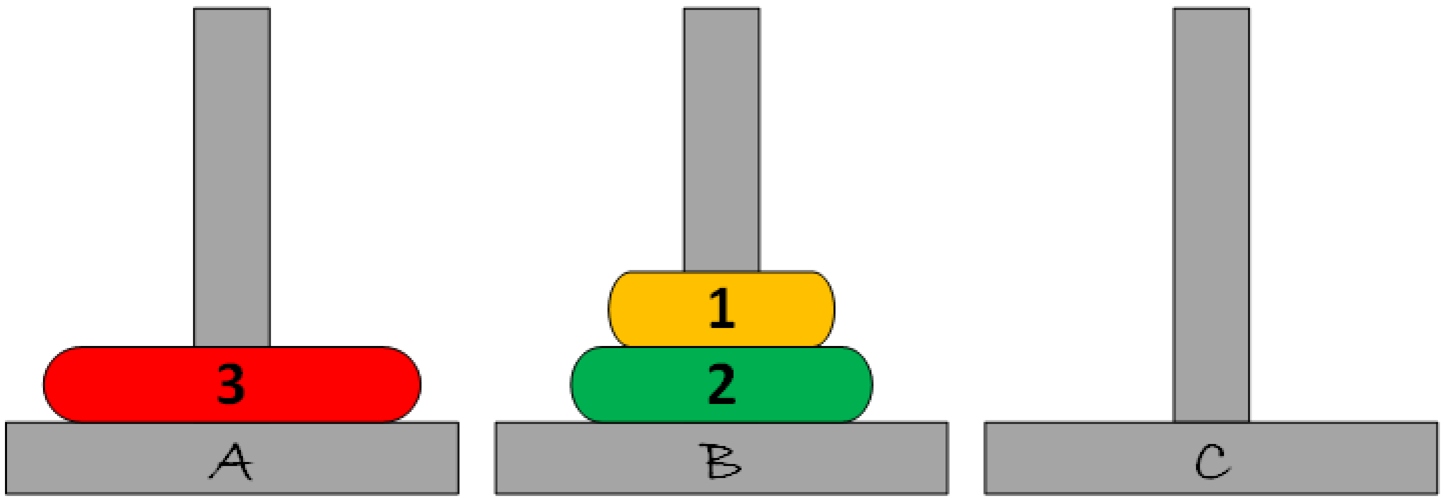

Step 1

# Example of Tower of Hanoi



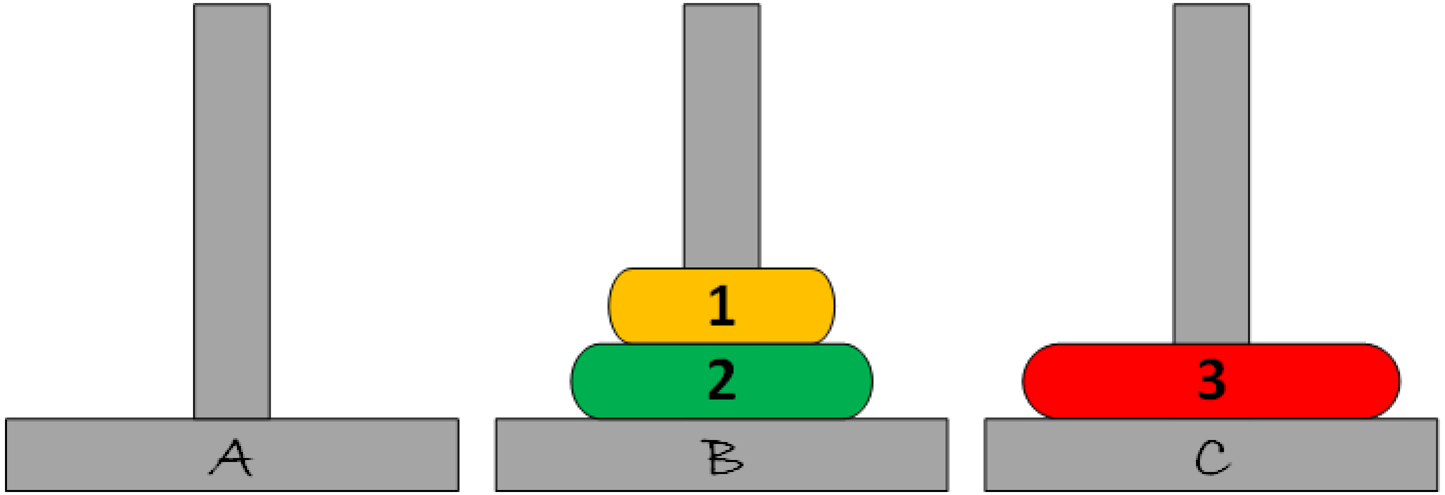

Step 2

# Example of Tower of Hanoi



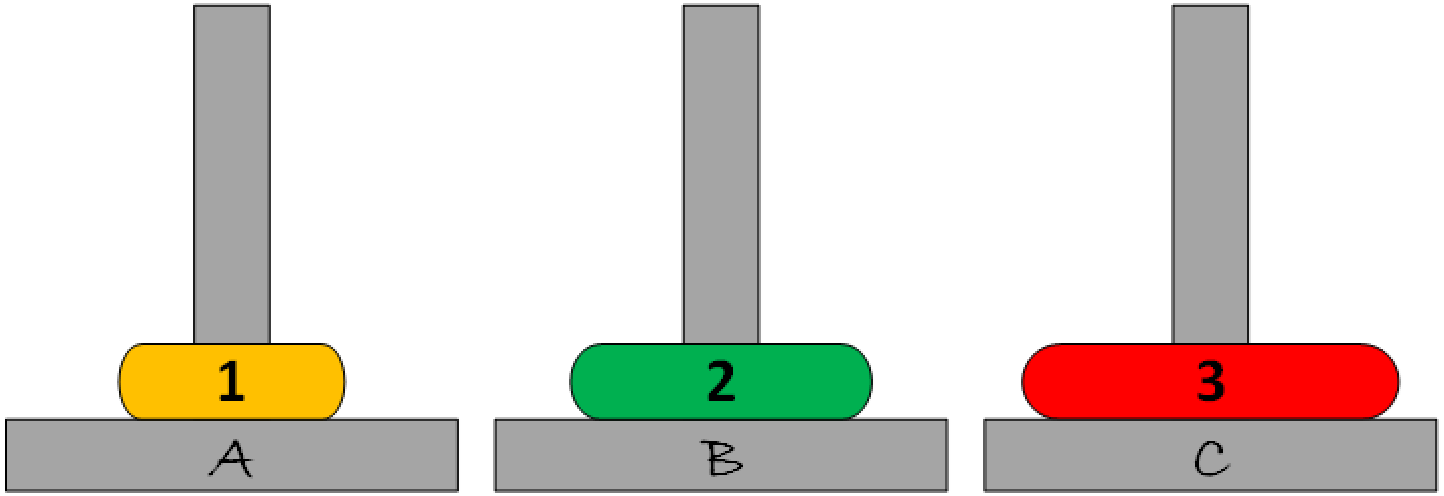

Step 3

# Example of Tower of Hanoi



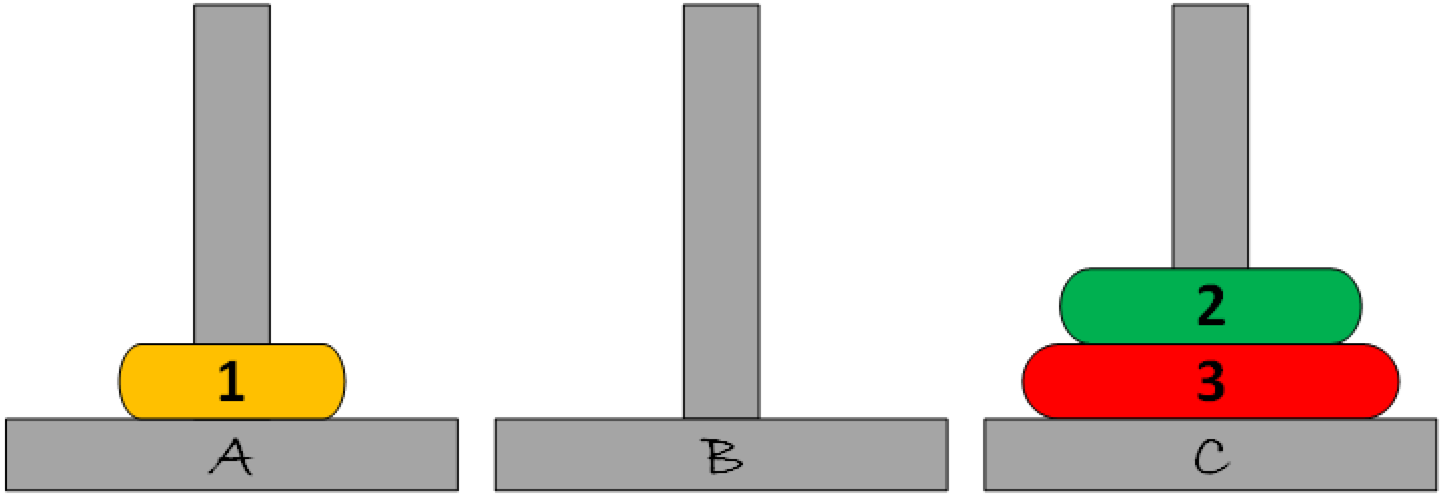

Step 4

# Example of Tower of Hanoi



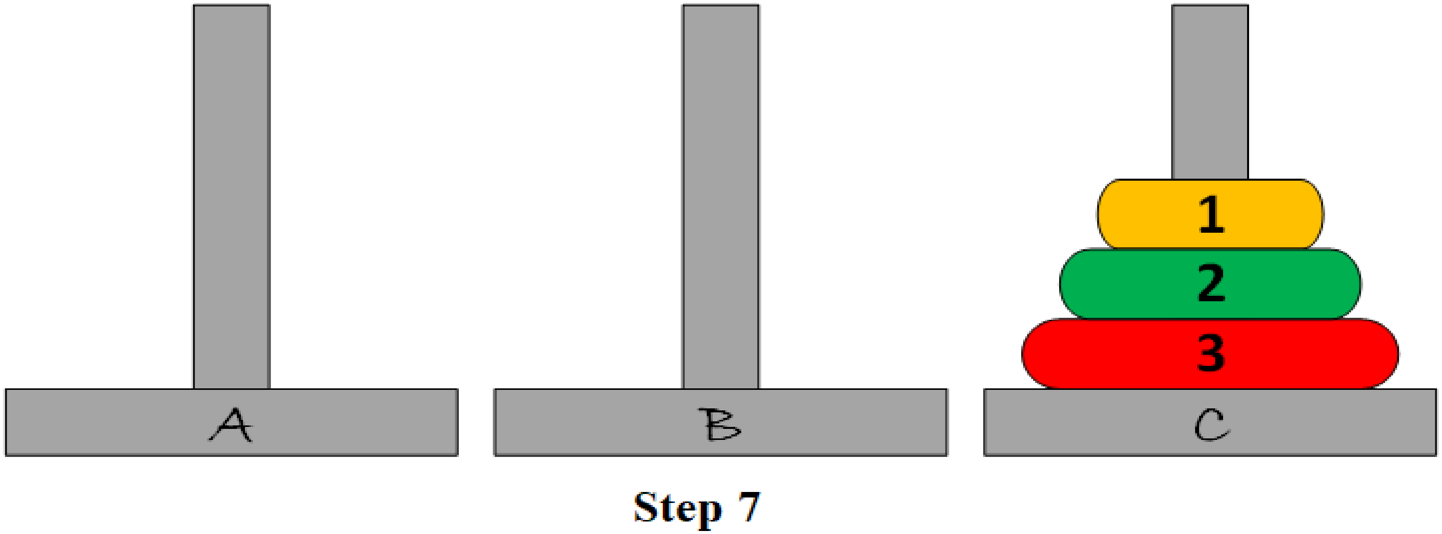

Step 5

# Example of Tower of Hanoi




Step 6

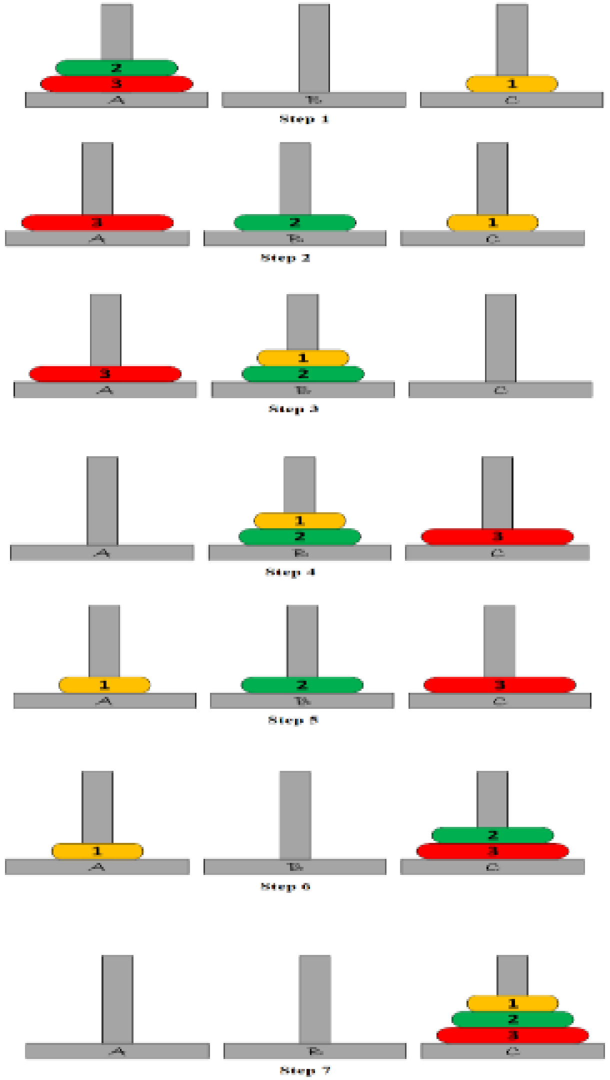
# Example of Tower of Hanoi




Animation with n=4:
https://www.openbookproject.net/py4fun/hanoi/hanoi3.html

# Example of Tower of Hanoi

- One possible solution (path from root to goal) is shown on the right.
  - **If a computer program can give you the steps, is that program "Intelligent"?**
  - The program is not intelligent, but it is programmed in a way that makes it look intelligent. The core algorithm is only listing all possible combinations and eliminating states that do not satisfy the input constraints.
- Attempt the problem if we have 7 disks instead of 3 (i.e., ***n*** =7).



# Recursive Functions

- A function defined in terms of itself
- **Example**: Factorial Function (n!)
- f(n) = n! = n x (n-1)! = n x f(n-1)
- In Programming (Python),
  a recursive function is a function
  that calls itself
- Creating such functions requires:
  1. Basis step
  2. Recursive Step

```python
# Non-Recursive definition
def fac1(n):
    f = 1
    for i in range(n, 1, -1):
        f = f * i
    return f

# Recursive definition
def fac2(n):
    if n == 1 :
        return 1                 # basis step
    else:
        return n * fac2(n - 1)   # Recursive step
```

# Example of Tower of Hanoi

*# Recursive Python function to solve the tower of Hanoi*

```
def TowerOfHanoi(n, source, destination, auxiliary):
    if n==1:
        print ("Move disk 1 from source", source, "to destination
              ",destination)
        return
    TowerOfHanoi(n-1, source, auxiliary, destination)
    print ("Move disk", n, "from source", source,
          "to destination", destination)
    TowerOfHanoi(n-1, auxiliary, destination, source)
```

*# Driver code*

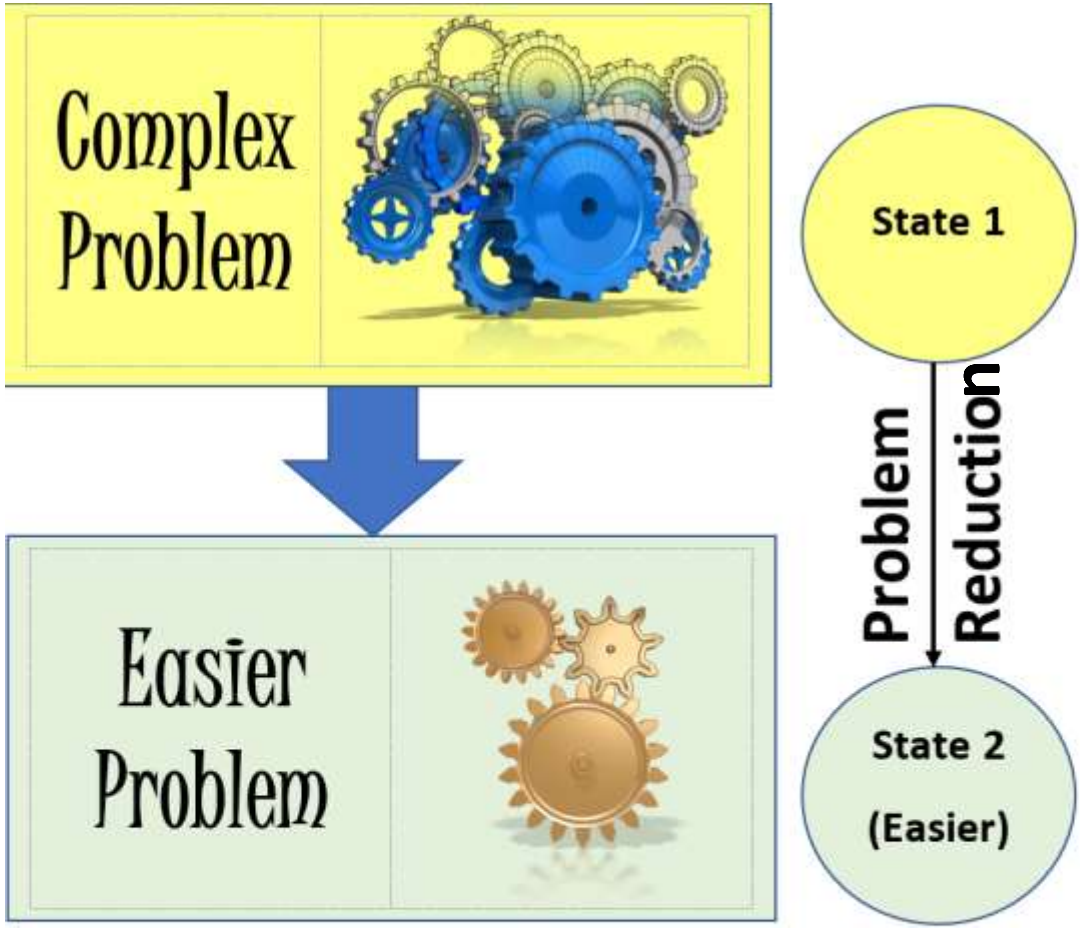
```
n = 3
TowerOfHanoi(n,'A','C','B')
```

*# A, C, B are the names of rods*

```
Move disk 1 from source A to destination C
Move disk 2 from source A to destination B
Move disk 1 from source C to destination B
Move disk 3 from source A to destination C
Move disk 1 from source B to destination A
Move disk 2 from source B to destination C
Move disk 1 from source A to destination C
```

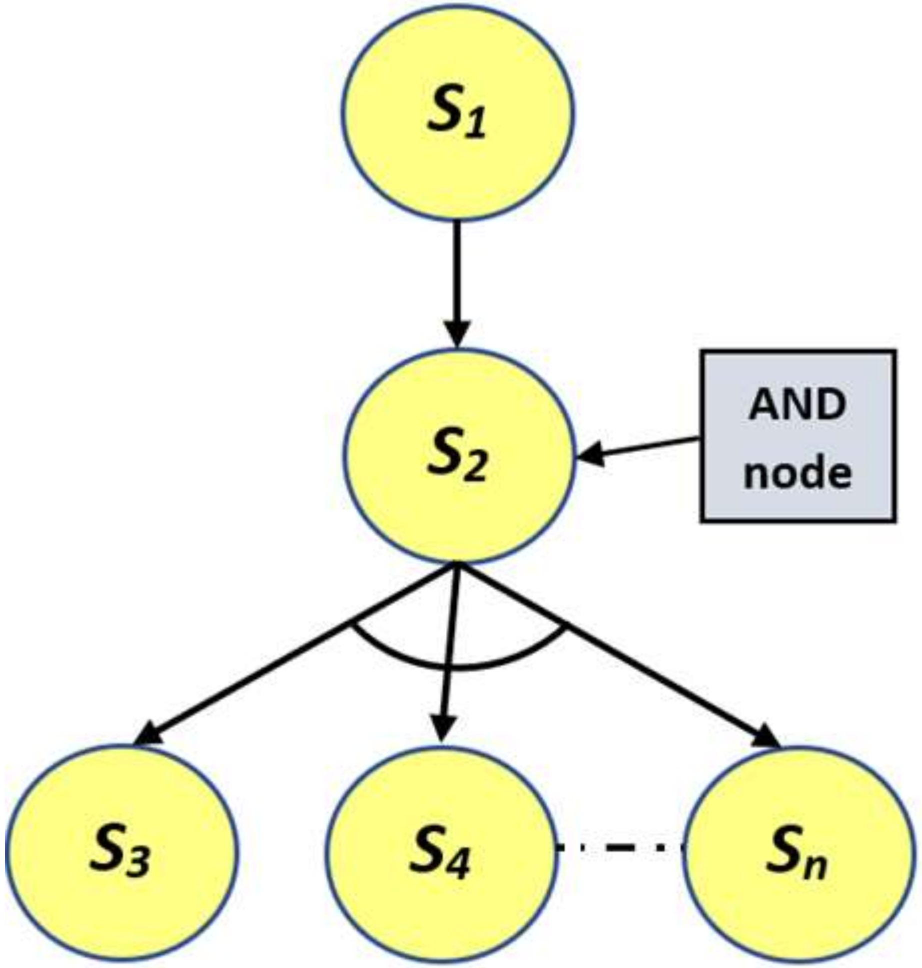
# Problem Reduction

- Sometimes we must divide a complex problem into different sets of easier independent problems
- In problem reduction, we will use the ***Generate and Test methodology*** where we will try to generate different sub-solutions (sub-goals) then test them to see if the goal is reached



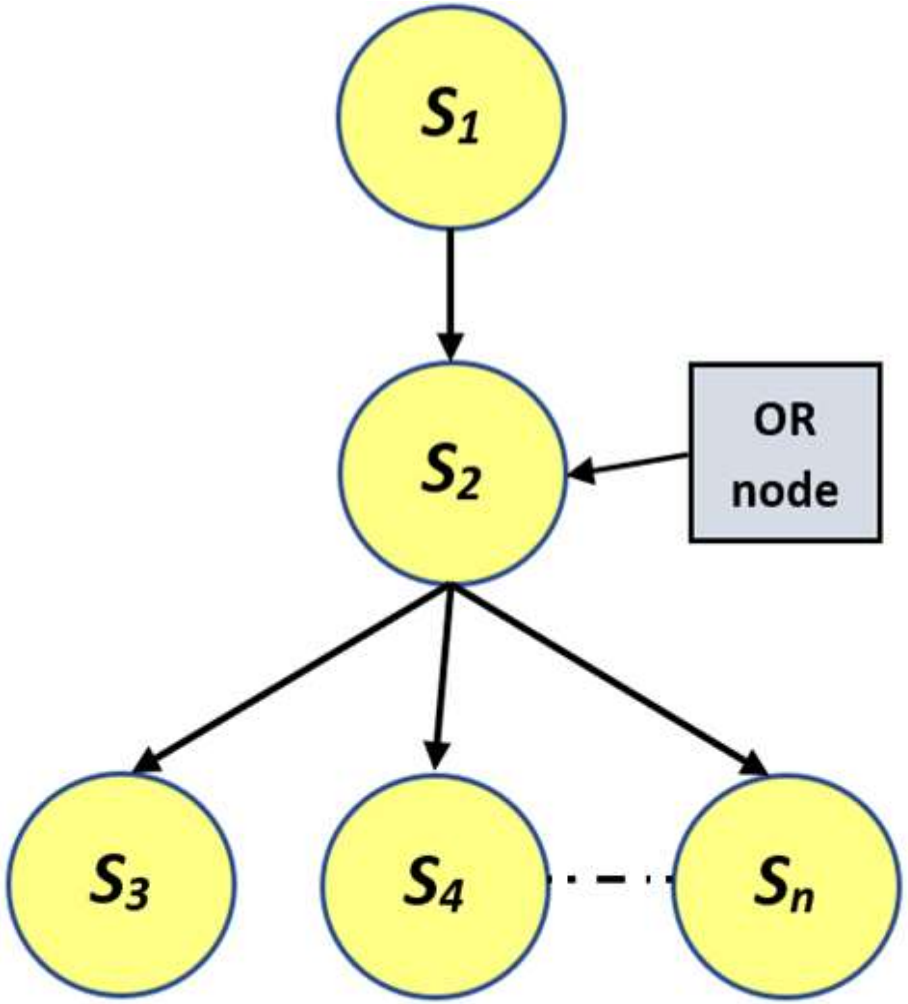
# Problem Reduction

- In problem reduction, we will use a **goal tree** which is a form of tree structure used to represent problems that can be broken down into smaller problems
  - Similar to that in Tower of Hanoi example
- The state at which a problem is broken into a set of independent problems is referred to as the **AND node**

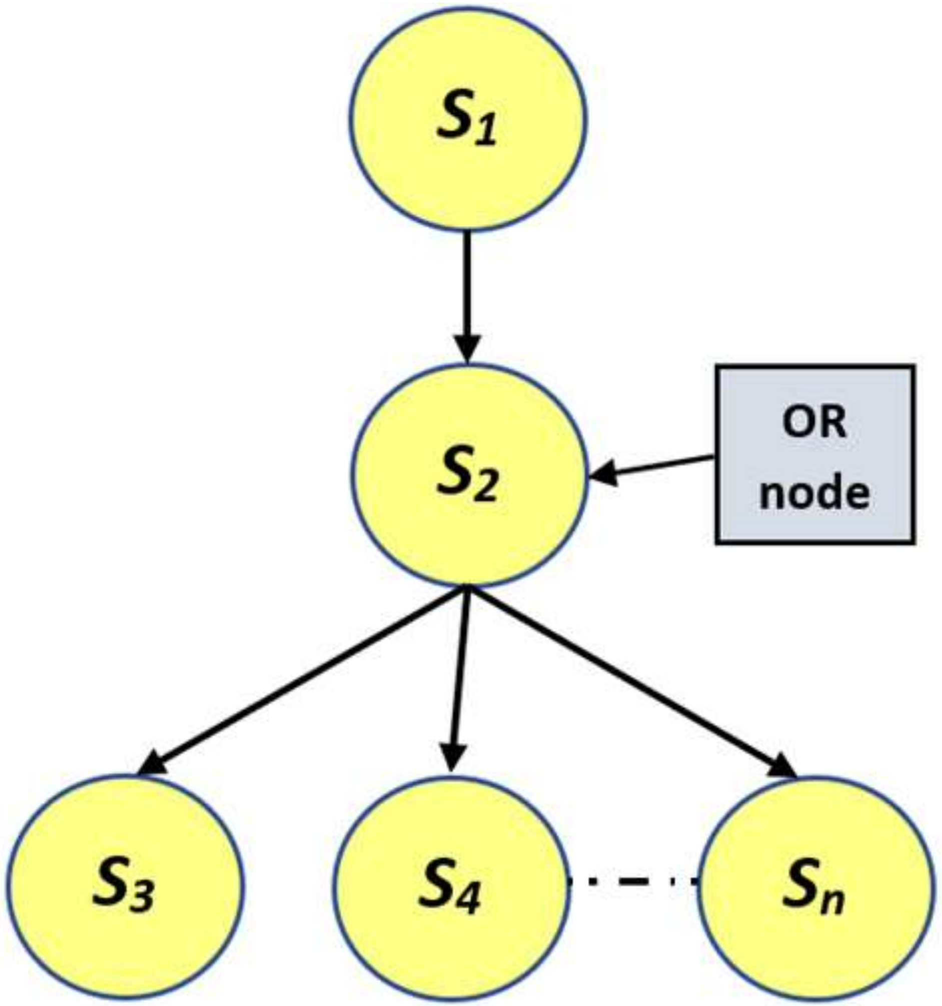


# Problem Reduction

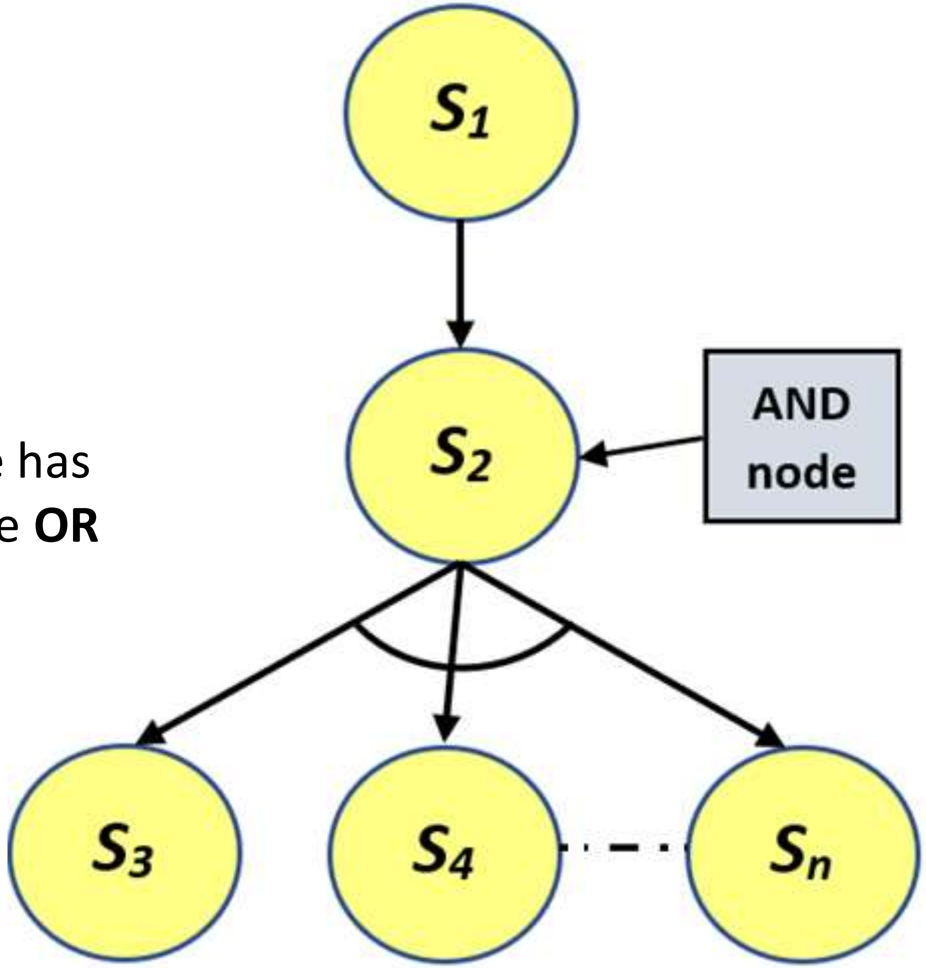
- Sometimes we have different ways to solve the problem, so we use **OR node** like the one shown
- Note that the **AND node** has the arch while the **OR node** does not



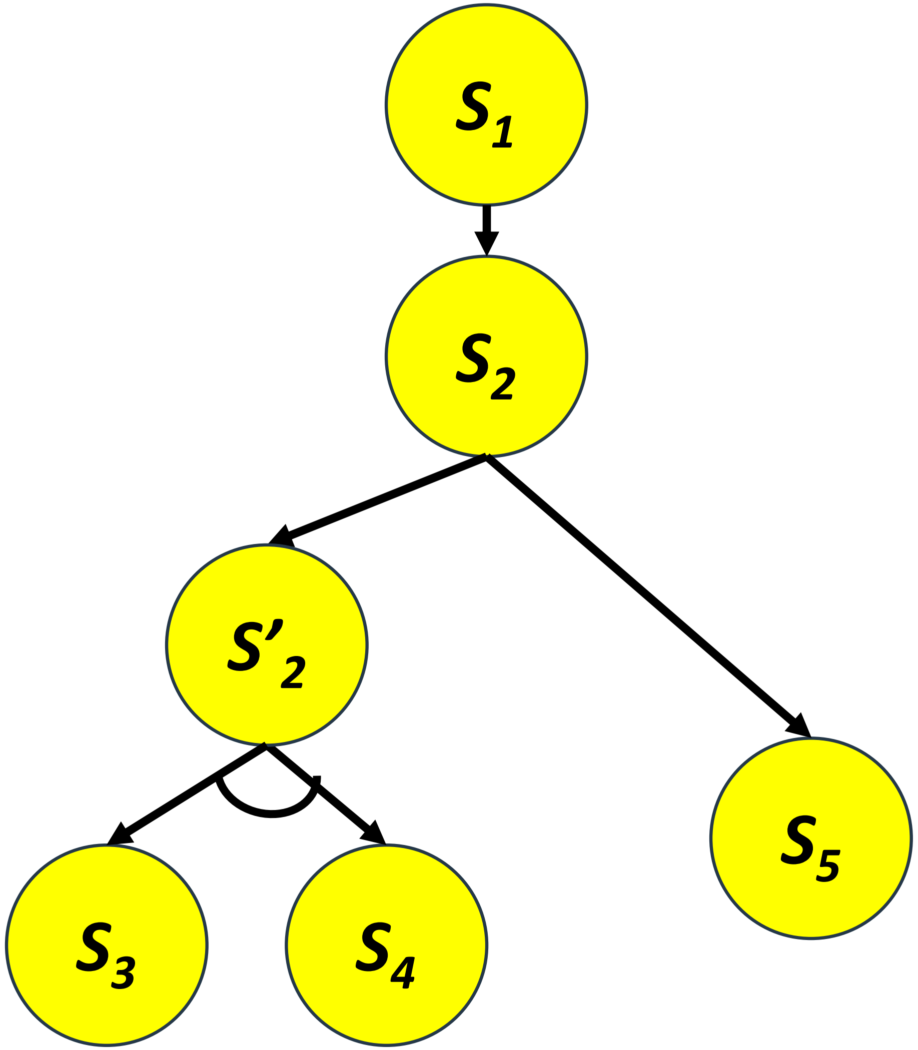
# AND node vs. OR node

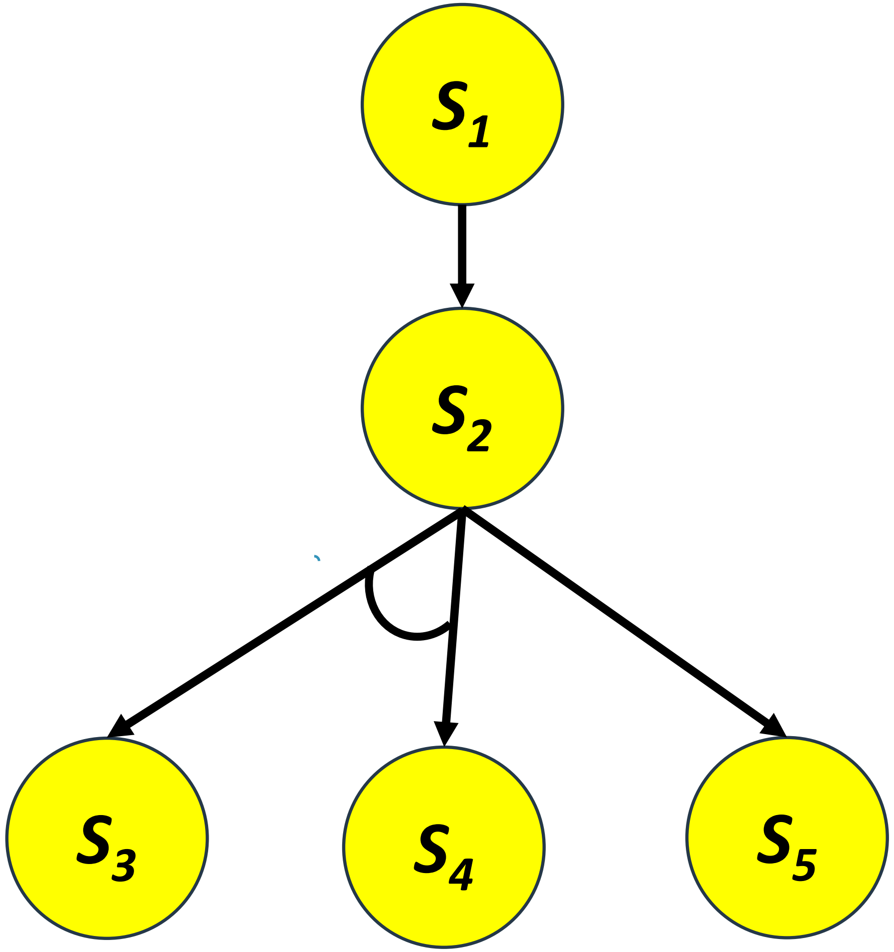



**Note**: the **AND** node has the arch whereas the **OR** node does not



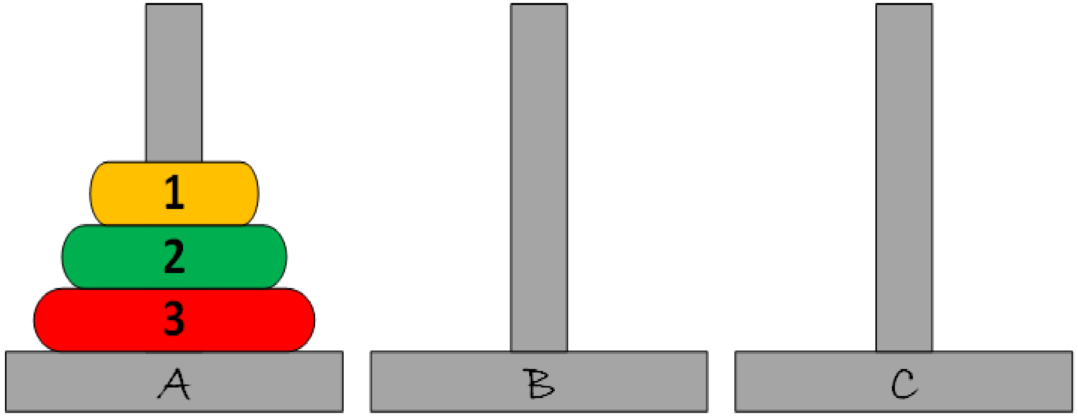
# AND-OR Shortcut



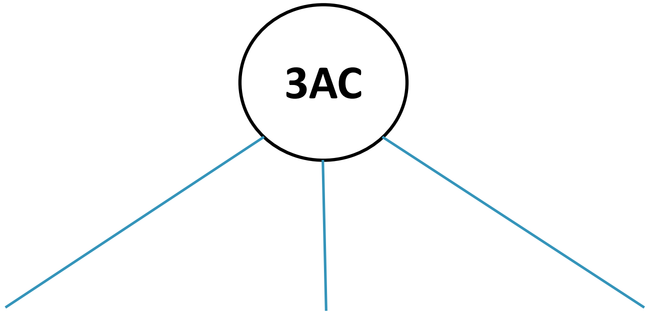



# Example of Tower of Hanoi – Problem Reduction

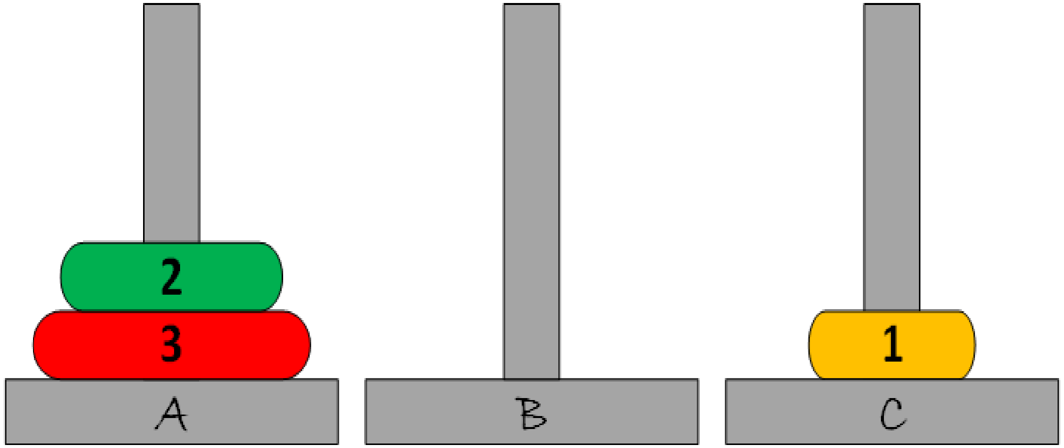
- The root node, labeled “3AC” represents the original problem of transferring all 3 disks from A to C
- The goal can be decomposed into three sub-goals: 2AB, 1AC, 2BC. In order to achieve the goal, all 3 sub-goals must be achieved



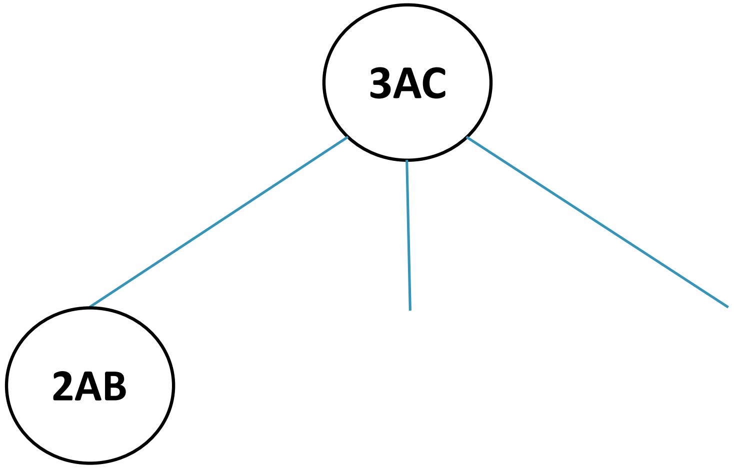

Initial State



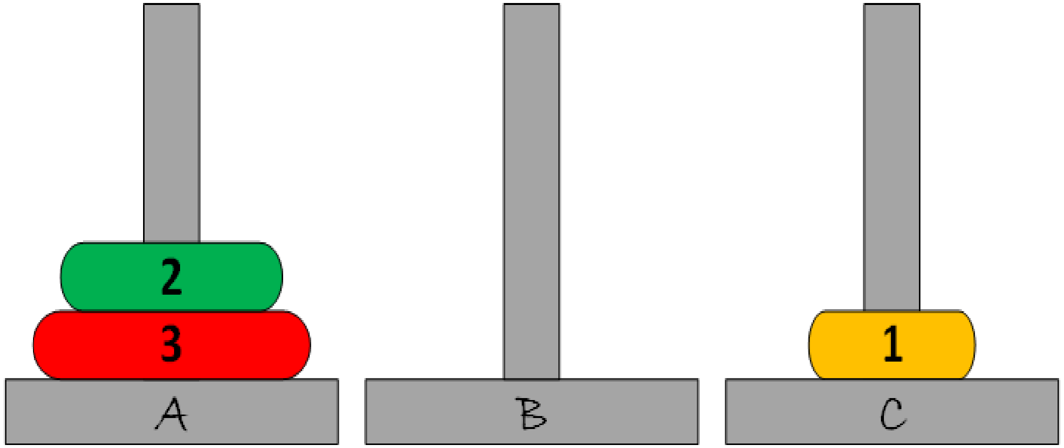
# Example of Tower of Hanoi – Problem Reduction



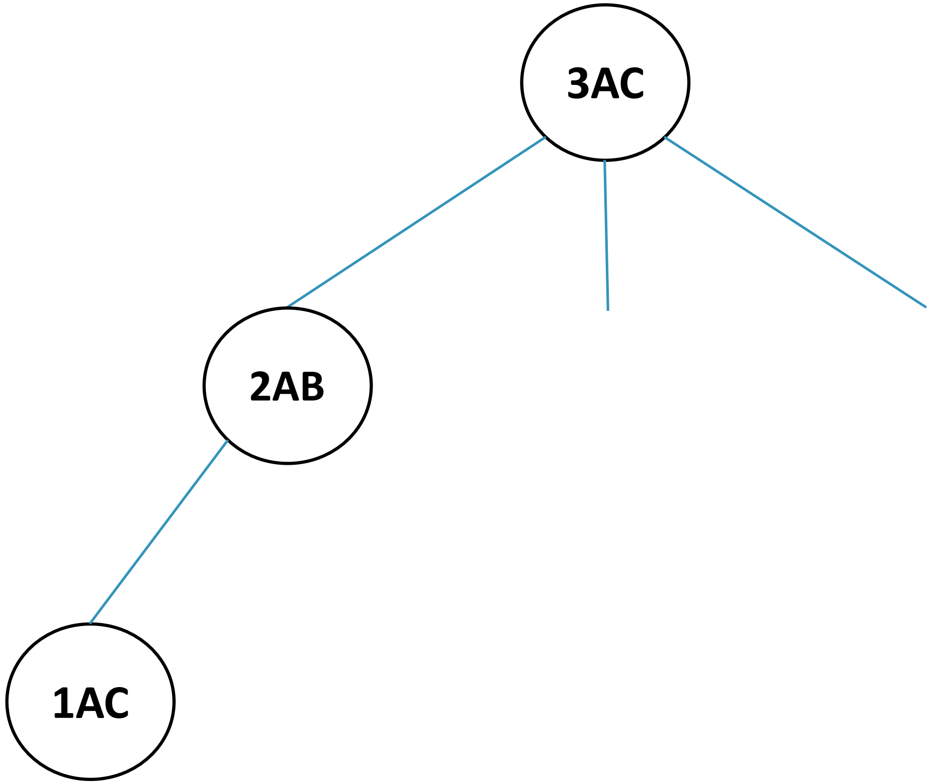

Step 1



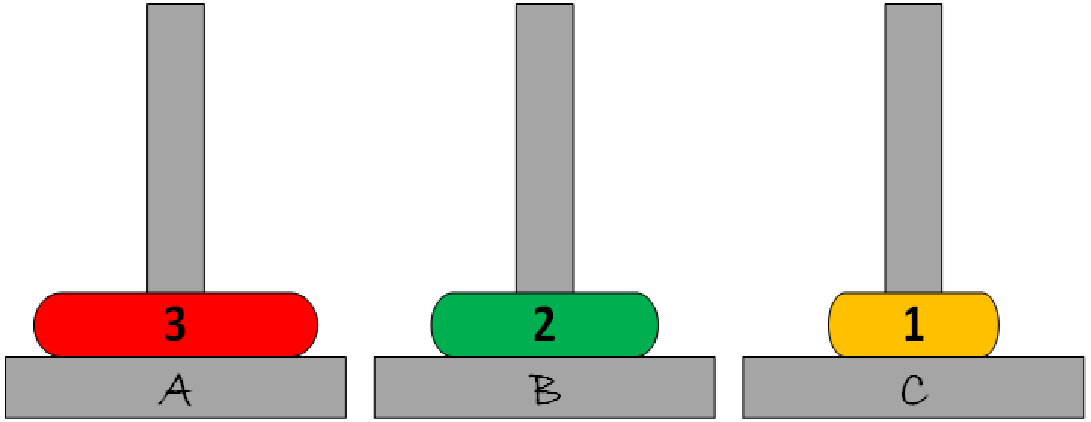
# Example of Tower of Hanoi – Problem Reduction



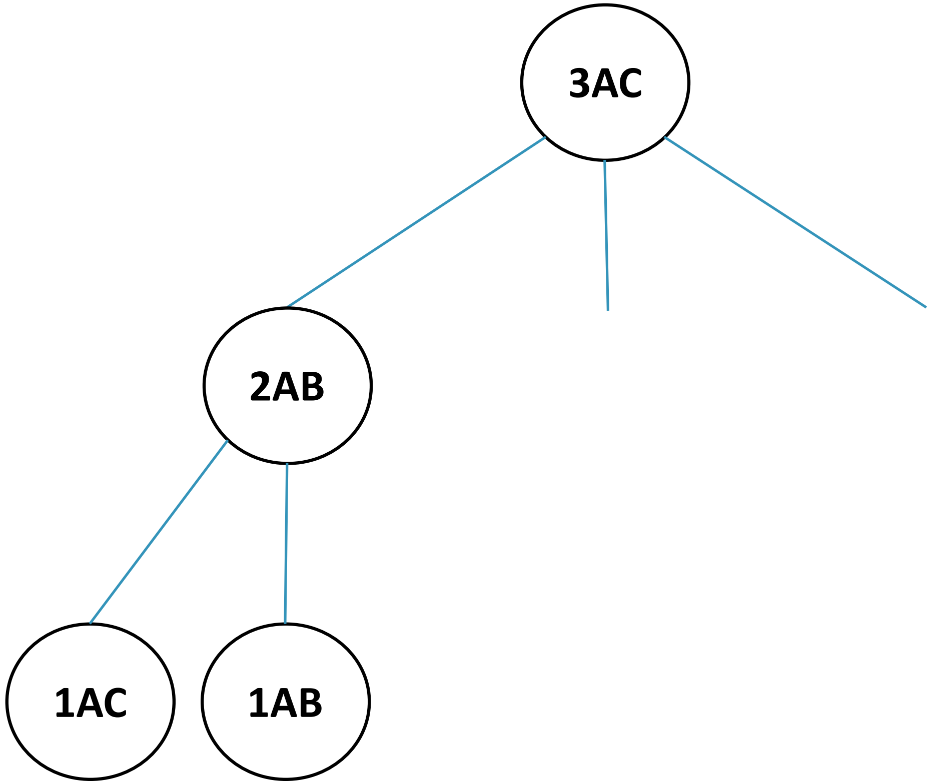

Step 1



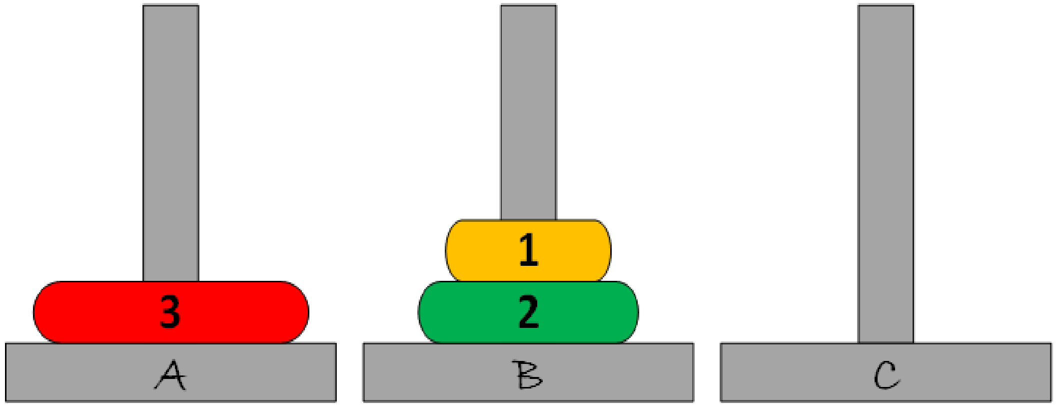
# Example of Tower of Hanoi – Problem Reduction



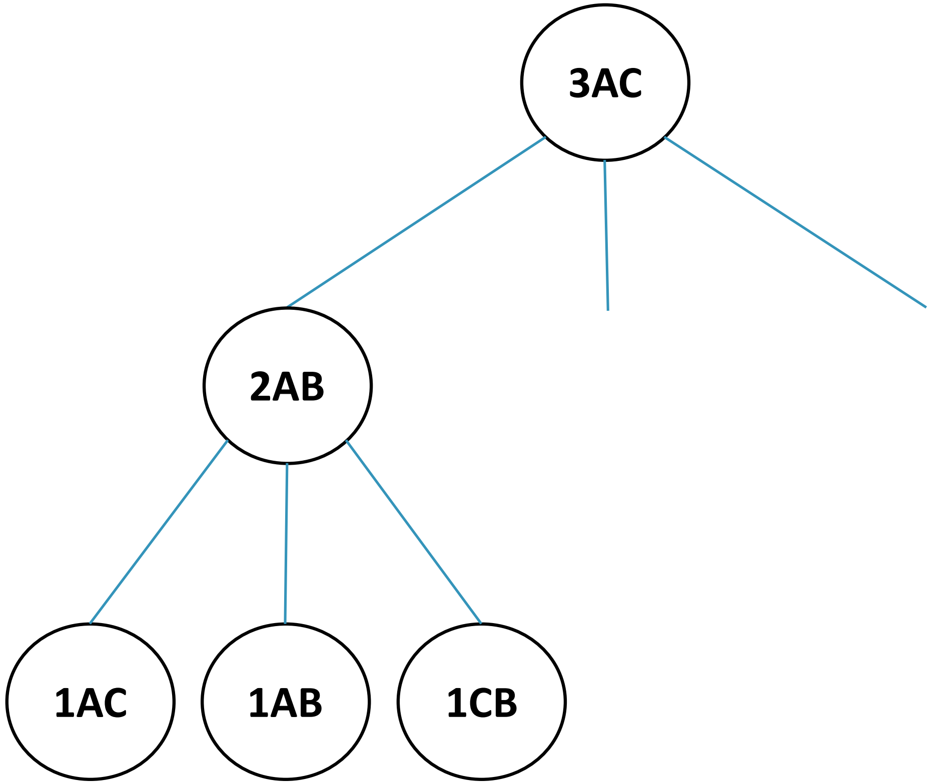

Step 2



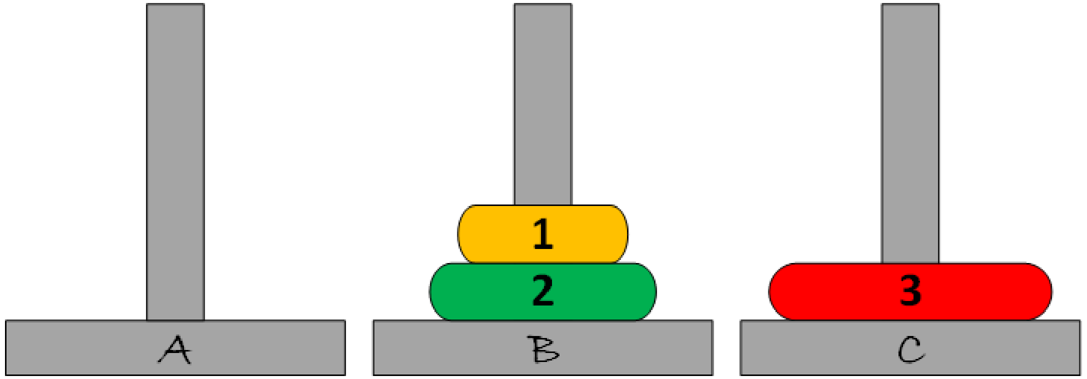
# Example of Tower of Hanoi – Problem Reduction



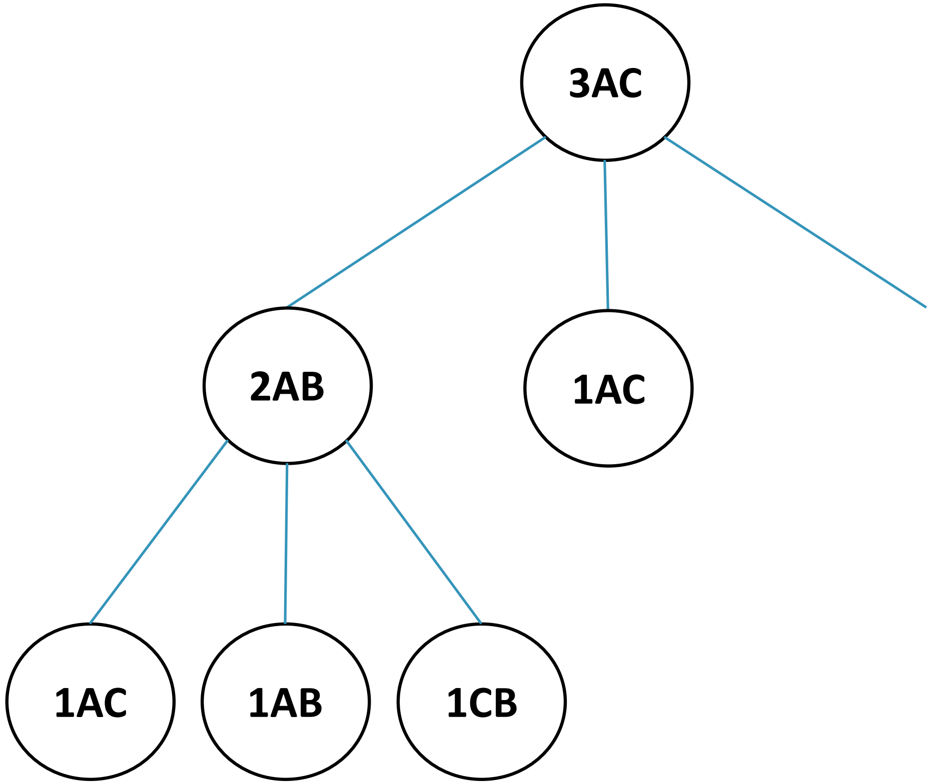

Step 3



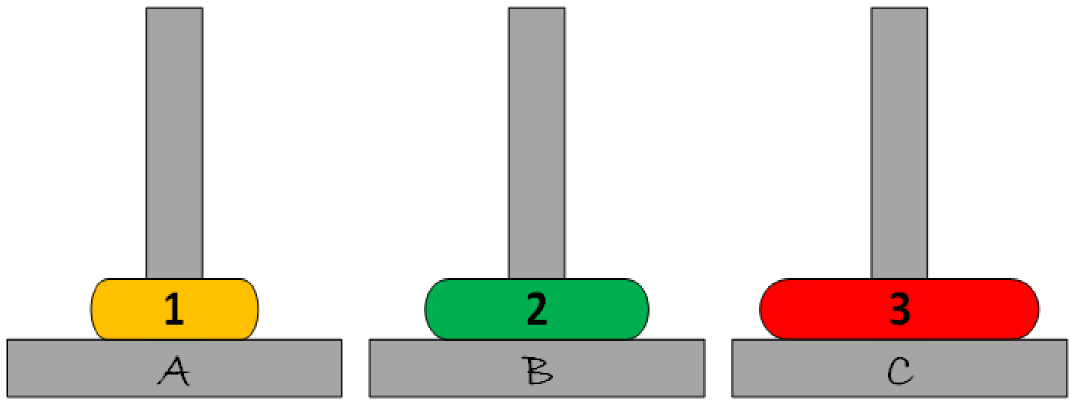
# Example of Tower of Hanoi – Problem Reduction



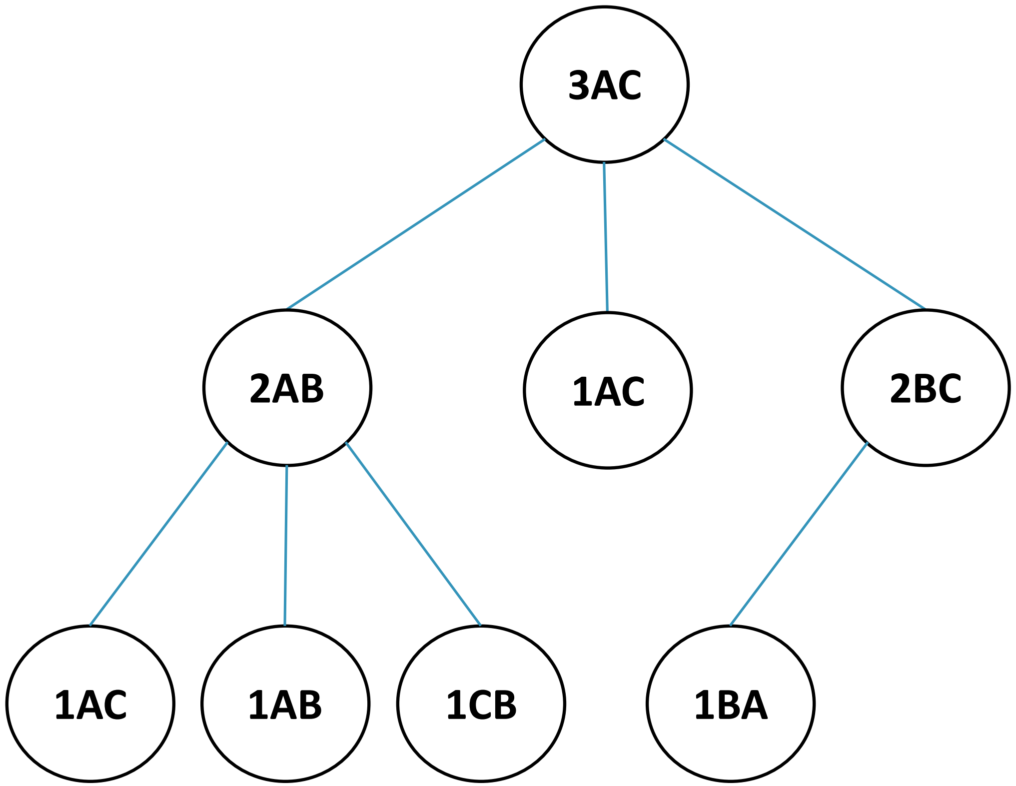

Step 4



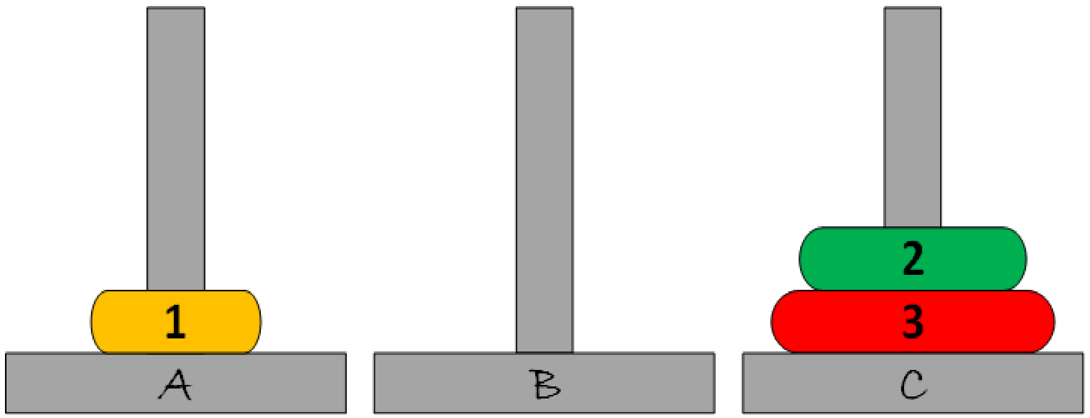
# Example of Tower of Hanoi – Problem Reduction



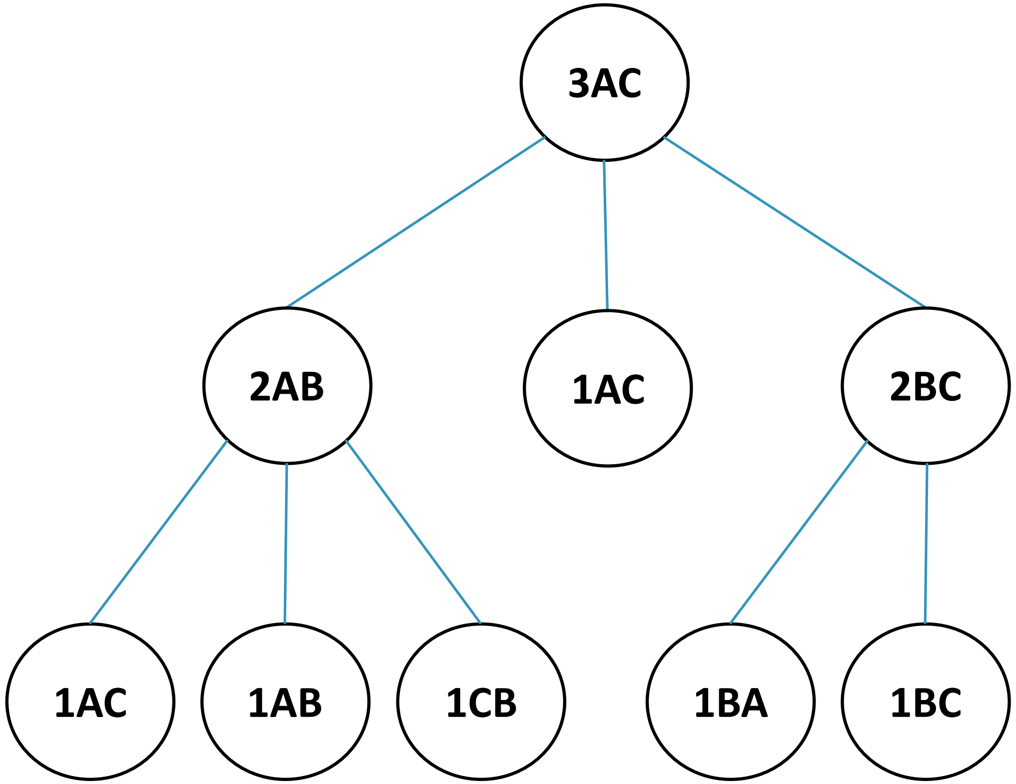

Step 5



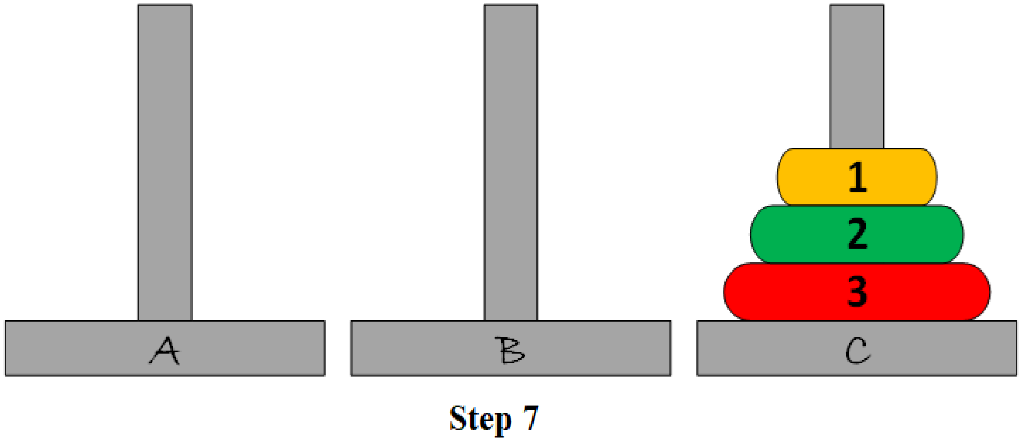
# Example of Tower of Hanoi – Problem Reduction

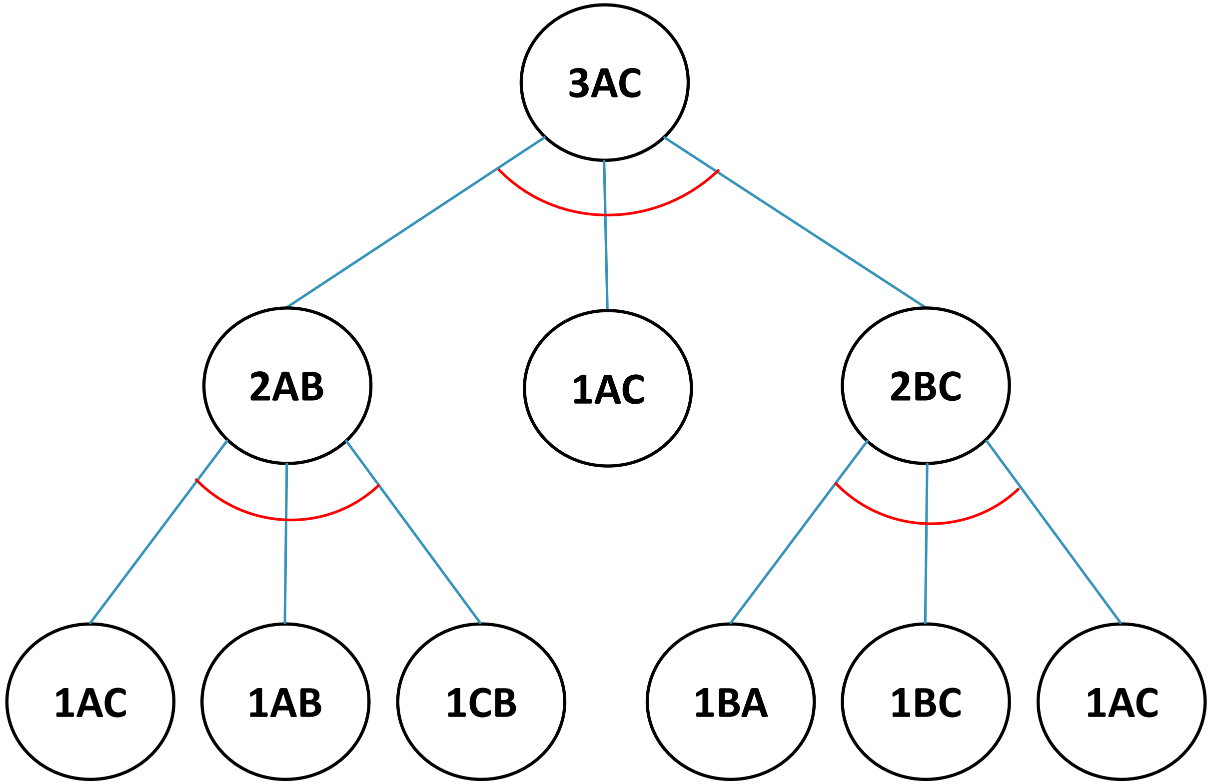



Step 6



# Example of Tower of Hanoi – Problem Reduction




https://www.youtube.com/watch?v=rf6uf3jNjbo

# Symbolic Integration using Problem Reduction

- The Start of AI was with Symbolic Integration
- Can you integrate the following equation: $\int \frac{-5x^4}{(1-x^2)^{5/2}} dx$
- Given that you have the integral table:

| # | Equation |
|---|---|
| a. | $\int \frac{1}{x} dx = \ln x$ |
| b. | $\int x^n \, dx = \frac{x^{n+1}}{n+1}$ |
| c. | $\int \cos x \, dx = \sin x$ |
| … | … |

# Symbolic Integration using Problem Reduction

- Modeling the way a human solves an integration problem
- Solution approach:

  Simplify the integral given to an easier problem that will help us in solving the integral
- Problem Reduction will utilize a known transform to reduce the original problem to a simpler one
- We will denote each step of problem reduction by a Node
- Review your calculus basics!

# Review of Integration Facts

- Safe Transformations: always good to do

| No | Name | Safe Transform |
|---|---|---|
| 1. | Sign Rule | $\int -f(x)dx = -\int f(x)dx$ |
| 2. | Constant Rule | $\int cf(x)dx = c\int f(x)dx$ |
| 3. | Sum Rule | $\int \sum f(x)\,dx = \sum \int f(x)dx$ |
| 4. | Division | $\int \frac{P(x)}{G(x)}$ ➔ Divide (if the degree of $P(x)$ is greater or equal to the degree of $G(x)$ perform long division |

Note that point 1 and 2 are the same but we will treat them differently for ease of understanding

# Review of Integration Facts

- Heuristic Transformations: sometimes useful, do not always work

| No | Name | Safe Transform |
|---|---|---|
| A. | Trigonometric | $f(\sin(x), \cos(x), \tan(x), \csc(x), \sec(x), \cot(x)) =$ $g(\sin(x), \cos(x)) =$ $g_2(\tan(x), \csc(x)) =$ $g_3(\cot(x), \sec(x))$ |
| B. | Trigonometric to polynomial | $\int f(\tan(x))dx = \int \frac{f(y)}{1+y^2} dy$ |
| C. | Polynomial to trigonometric | If you see the term $1 - x^2$ in an integral do a substitution of $x = \sin(y)$ and $dx = \cos y dy$ |
| D. | Polynomial to trigonometric | If you see the term $1 + x^2$ in an integral do a substitution of $x = \tan(y)$ and $dx = sec^2 y dy$ |

# Trigonometric Identities

**Pythagorean identities :**

$$\sin^2\theta + \cos^2\theta = 1$$

$$\tan^2\theta + 1 = \sec^2\theta$$

$$1 + \cot^2\theta = \csc^2\theta$$

**Reciprocal identities :**

$$\csc x = \frac{1}{\sin x}$$

$$\sec x = \frac{1}{\cos x}$$

$$\cot x = \frac{1}{\tan x}$$

**Product to sum formulas :**

$$\sin x \cdot \sin y = \frac{1}{2}\left[\cos(x-y) - \cos(x+y)\right]$$

$$\cos x \cdot \cos y = \frac{1}{2}\left[\cos(x-y) + \cos(x+y)\right]$$

$$\sin x \cdot \cos y = \frac{1}{2}\left[\sin(x+y) + \sin(x-y)\right]$$

$$\cos x \cdot \sin y = \frac{1}{2}\left[\sin(x+y) - \sin(x-y)\right]$$

**Sum and difference formulas :**

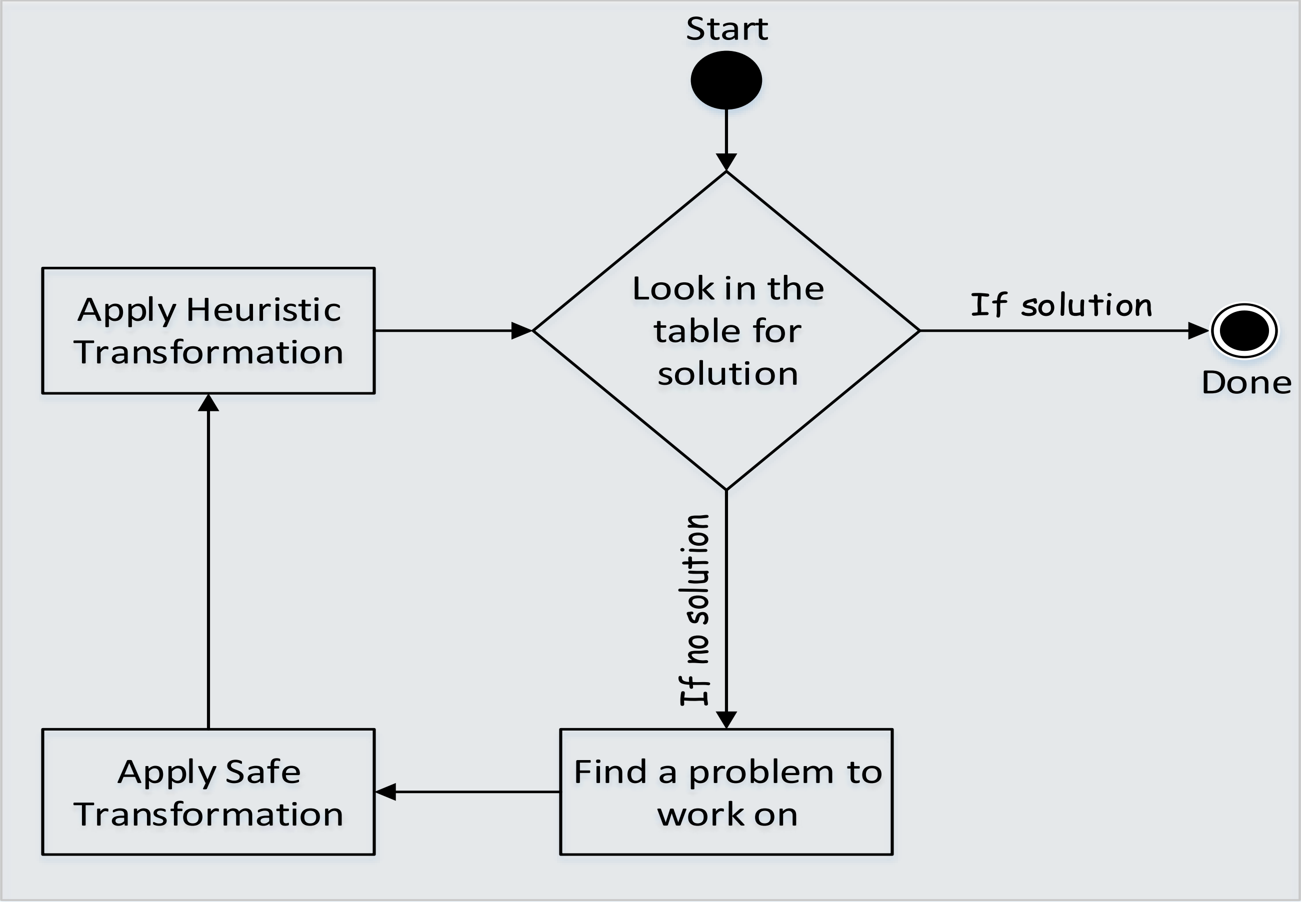
$$\sin(x \pm y) = \sin x \cos y \pm \cos x \sin y$$

$$\cos(x \pm y) = \cos x \cos y \mp \sin x \sin y$$

$$\tan(x \pm y) = \frac{\tan x \pm \tan y}{1 \mp \tan x \tan y}$$

# Solving the Symbolic Integration Problem

- We will use Generate and Test methodology as shown below:



# Symbolic Integration Problem

- The original integration problem can be simplified to:

$$\int \frac{-5x^4}{(1-x^2)^{5/2}}\,dx \rightarrow \int \frac{5x^4}{(1-x^2)^{\frac{5}{2}}}\,dx \rightarrow \int \frac{x^4}{(1-x^2)^{5/2}}\,dx$$

- Using the safe transforms

$$\int -f(x)dx = -\int f(x)dx\,, \int cf(x)dx = c\int f(x)dx$$

- The reduced problem is simpler now but still significantly hard, need more transformations

# Symbolic Integration Problem

- By substituting Heuristic Transformation (C) and (A)

$$x = \sin y$$

$$dx = \cos y dy$$

$$f(\sin(x), \cos(x), \tan(x), \csc(x), \sec(x), \cot(x)) = g(\sin(x), \cos(x)) = g(\tan(x), \csc(x))$$

- The problem can be simplified to:

$$\int \frac{x^4}{(1-x^2)^{5/2}} dx \to \int \frac{\sin^4 y}{(1-\sin^2 y)^{5/2}} \cos y dy \to \int \frac{\sin^4 y}{(cos^2 y)^{5/2}} \cos y \, dy \to \int \frac{\sin^4 y}{\cos^4 y} dy$$

- The integration $\int \frac{\sin^4 y}{\cos^4 y} dy$ can be reduced to solving

1. Either $\int \frac{1}{\cot^4 y} dy$
2. Or $\int \tan^4 y \, dy$

which one will you choose?

# Symbolic Integration Problem

- Using the Heuristic Transformation (B) we get

$$\int f(\tan(x))dx = \int \frac{f(y)}{1+y^2}dy$$

- and substituting $(z = \tan(y))$ we can write

$$\int \tan^4 y dy \rightarrow \int \frac{z^4}{1+z^2}dz$$

- Since the power of the numerator is larger than the power of the denominator, we can simplify by doing long division (safe transformation 4), thus the equation is further simplified to:

$$\int \frac{z^4}{1+z^2}dz \rightarrow \int \left(z^2 - 1 + \frac{1}{1+z^2}\right)dz$$

# Symbolic Integration Problem

- Using Safe Transformation 3 (Integral Sum Rule) we can write:

$$\int \left(z^2 - 1 + \frac{1}{1+z^2}\right) dz \rightarrow \int z^2 dz - \int dz + \int \frac{1}{1+z^2} dz$$

- The first and second term can be found using:

$$\int x^n dx = \frac{x^{n+1}}{n+1}$$

- Let us concentrate on the third term:

$$\int \frac{1}{1+z^2} dz$$

# Symbolic Integration Problem

- By doing the following substitution

$$z = \tan w \rightarrow dz = \frac{1}{\cos^2 w} dw$$

- We can write:

$$\int \frac{1}{1+z^2} dz \rightarrow \int \frac{1}{1+\tan^2 w} \times \frac{1}{\cos^2 w} dw \rightarrow \int \frac{1}{\sec^2 w} \sec^2 w dw \rightarrow \int dw$$

- The above can be solved using

$$\int x^n dx = \frac{x^{n+1}}{n+1} \blacksquare$$

# Symbolic Integration Problem (Summary)

A: $\int \frac{-5x^4}{(1-x^2)^{5/2}} dx \rightarrow$ B: $\int \frac{5x^4}{(1-x^2)^{5/2}} dx \rightarrow$ C: $\int \frac{x^4}{(1-x^2)^{5/2}} dx \rightarrow$ D: $\int \frac{\sin^4 y}{(1-\sin^2 y)^{5/2}} \cos y dy$

E: $\int \frac{\sin^4 y}{\cos^4 y} dy$

F: $\int \frac{1}{\cot^4 y} dy$

G: $\int \tan^4 y \, dy \rightarrow$ H: $\int \frac{z^4}{1+z^2} dz$

I: $\int \left(z^2 - 1 + \frac{1}{1+z^2}\right) dz$

J: $\int z^2 dz$

K: $\int -1 \, dz \rightarrow$ L: $\int dz$

M: $\int \frac{1}{1+z^2} dz \rightarrow$ N: $\int dw$

# Symbolic Integration Problem - Goal Trees

- Rearranging nodes with symbols we get

$$\text{A: } \int \frac{-5x^4}{(1-x^2)^{5/2}}dx \rightarrow \text{B: } \int \frac{5x^4}{(1-x^2)^{5/2}}dx \rightarrow \text{C: } \int \frac{x^4}{(1-x^2)^{5/2}}dx \rightarrow \text{D: } \int \frac{\sin^4 y}{(1-\sin^2 y)^{5/2}}\cos y\,dy$$

$$\rightarrow \text{E: } \int \frac{\sin^4 y}{\cos^4 y}dy$$

$$\text{E} \rightarrow \text{F: } \int \frac{1}{\cot^4 y}dy$$

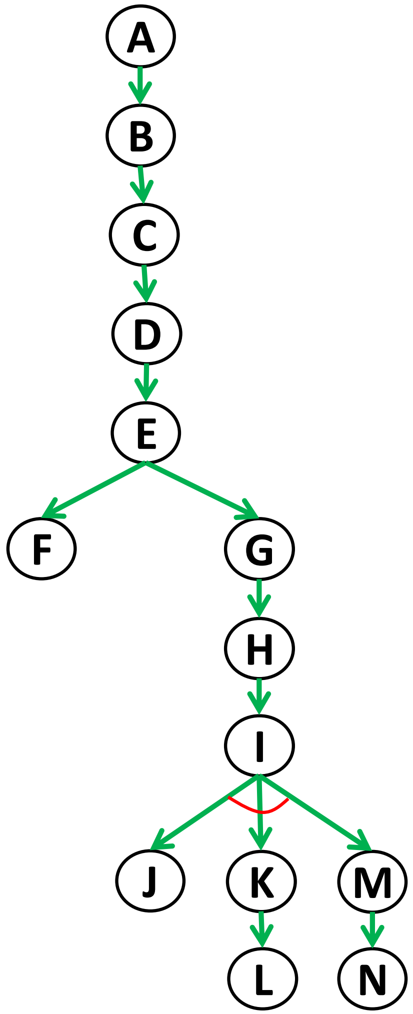
$$\text{E} \rightarrow \text{G: } \int \tan^4 y\,dy \rightarrow \text{H: } \int \frac{z^4}{1+z^2}dz$$

$$\rightarrow \text{I: } \int \left(z^2 - 1 + \frac{1}{1+z^2}\right)dz$$

$$\text{I} \rightarrow \text{J: } \int z^2 dz$$

$$\text{I} \rightarrow \text{K: } \int -1\,dz \rightarrow \text{L: } \int dz$$

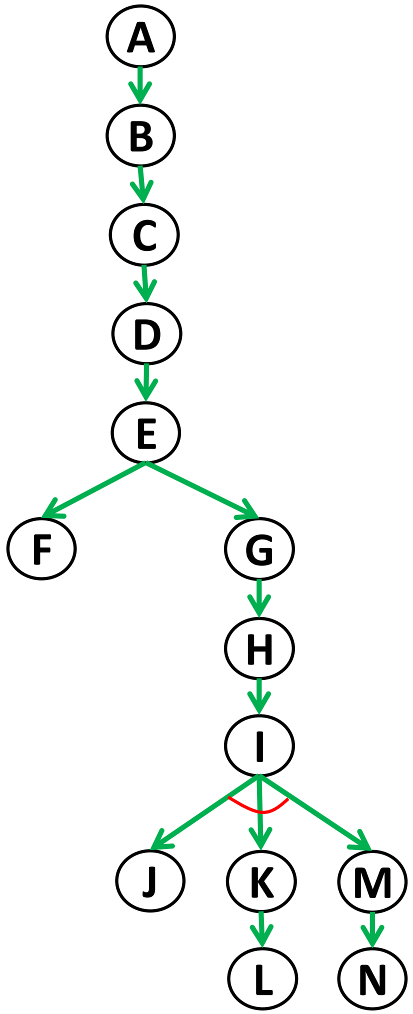
$$\text{I} \rightarrow \text{M: } \int \frac{1}{1+z^2}dz \rightarrow \text{N: } \int dw$$




- A *Goal Tree*

# Understanding Goal Trees

- General Tree structure
  - A tree has nodes and edges where edges connect two nodes within the tree
  - A branch appears when the node has more than one edge
  - A tree depth is the number of levels in the goal tree to reach to the solution
- The tree depth for the integration problem is 9

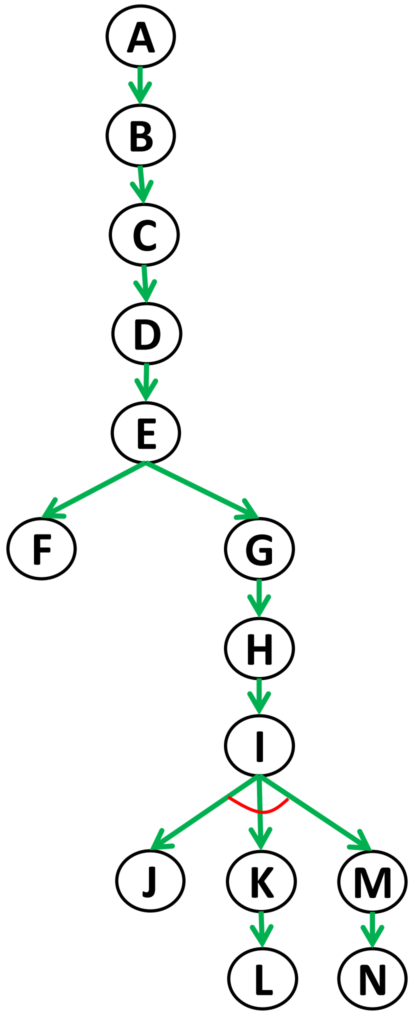


# Understanding Goal Trees

- Tree depth gives a lot of information such as:
  - Complexity of the problem: As the tree depth grows, the problem is viewed as being more complex since it needs more steps to solve the problem
  - The complexity can be used to identify the level of intelligence. An entity (such as a person, computer, etc.) that can solve a problem with a higher tree depth is more intelligent than programs that solve less depth problems
  - Note that if multiple solutions of a problem exist with different tree depth, the smart choice is to use the shortest tree

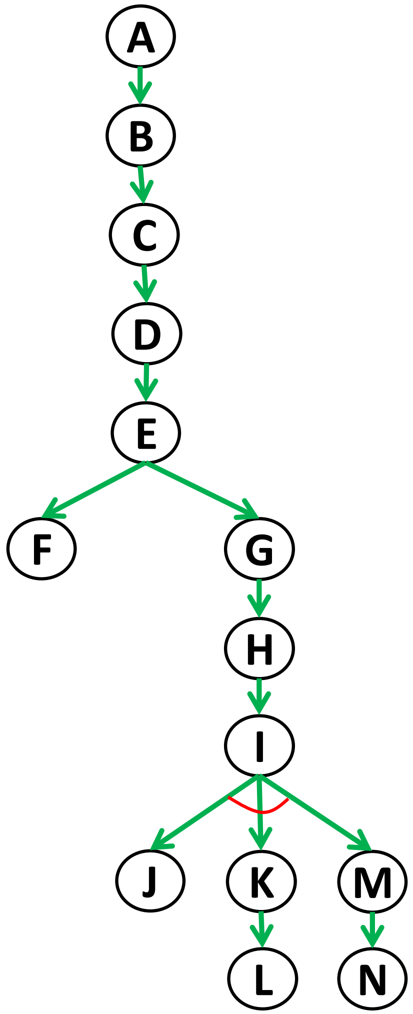
# Understanding Goal Trees

- Goal trees can help us answer "How" and "Why" questions
  - The answer to "How" question is going down in the tree level by level
  - The answer to "Why" question is going up in the tree by one level



# Understanding Goal Trees

- In the integration program, the following was observed:
  - The program was given 56 freshman-level problems from which it solved 54 difficult problems and failed in two
  - The worst-case tree depth of the 56 problems was 7 (taking away the -5; our example)
  - Average tree depth for solving integration problems was 3
  - Number of unused branches is 1 (i.e., did not need to solve it to produce a solution (just like node ***F*** in the tree)



# Strategy to Solve Symbolic Integration Summary

- Start with safe transformations, the ones you are sure will work in any case. Then apply heuristic transformations, the ones that could work
- The problem simplification schema, may create
  - "**and node**" where the problem forks into several sub problems and
  - "**or node**" where the problem may be solved with either one or another transformation
- The resulting schema is usually called a "problem reduction tree", "and/or tree" or "goal tree"
- Note: In an "**or node**", it helps to understand the depth of functional composition (number of transformations to be applied after the "or" options of the branch) and the simplicity of solving each option to complete the problem resolution

# To get the Goal Tree of a Certain Problem

1. Start by evaluating what kind of knowledge is involved [Integration Tables; Transformations; Goal Tree]
2. Understand how the knowledge is represented. Each category of knowledge has its own way of being represented. [Expressions; Tables; Goal Trees as Procedures]
3. Know how the knowledge is used.[Transformations to make problems simpler; Integration Tables to trim bottom of Tree]
4. Know how much knowledge is required to solve the problem. [26 integration tables; 12 safe transformations; 12 heuristic transformations]

# Conclusion

- Many problems can be visually represented in form of goal trees
- Trees aid our understanding of the problem and show the difficulty of the problem by looking at the depth of the tree
- A Goal Tree program can answer questions about its own behavior by reporting steps up (why questions) or down (how questions) in the actions it takes