Lecture 4
Artificial Intelligence

Khola Naseem
khola.naseem@uet.edu.pk

Problem Solving by Searching

- Why search ?
- Early works of AI was mainly towards
  - proving theorems
  - solving puzzles
  - playing games
- All AI is search!
  - Not totally true (obviously) but more true than you might think.
- All life is problem solving !!
  - Finding a good/best solution to a problem amongst many possible solutions.

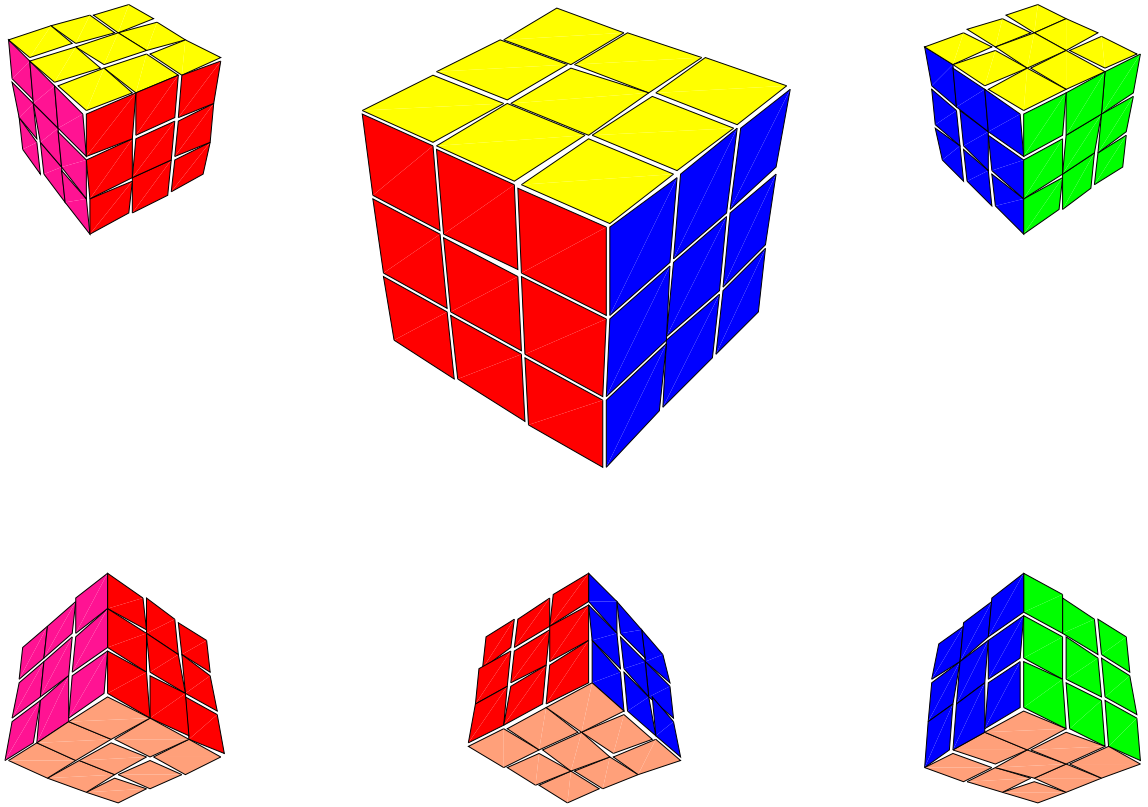
Credit: Khola Naseem

Problem Solving by Searching

- The simplest agents **Simple reflex agents**, which base their actions on a direct mapping from states to actions.
- Such agents cannot operate well in environments for which this mapping would be too large to store and would take too long to learn
- **Goal-based agents**, on the other hand, consider future actions and the desirability of their outcomes.
- Therefore, Goal-based agents is used

Classic AI Search Problems

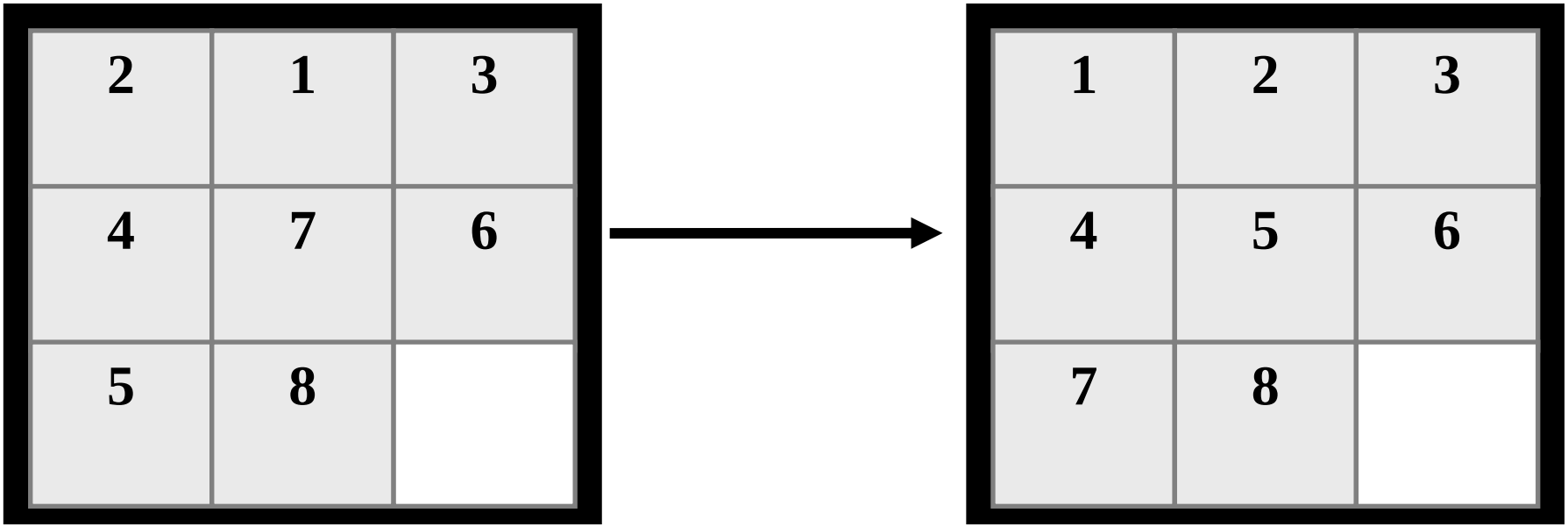
- 3*3*3 Rubik's Cube



Credit: Khola Naseem

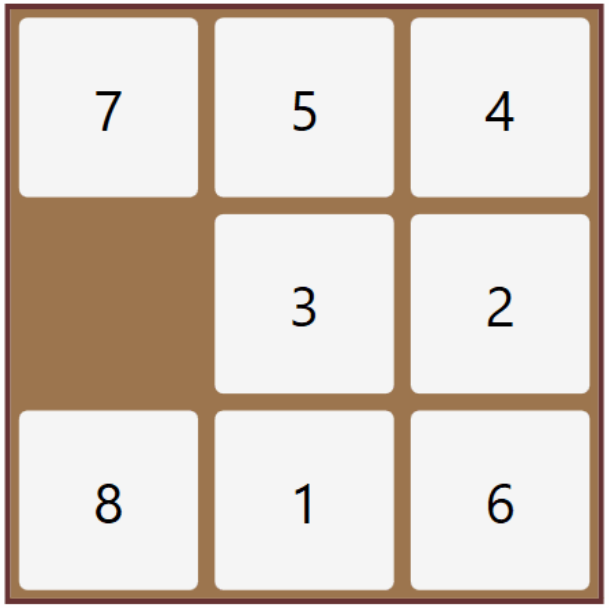
Classic AI Search Problems

- 8 Puzzle problem



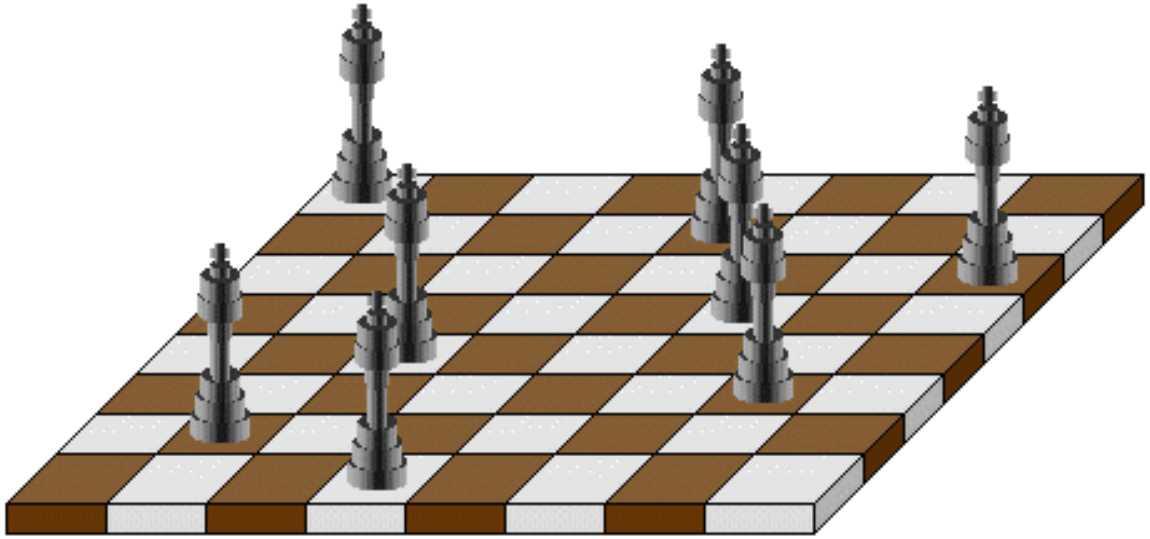
Classic AI Search Problems

- 8 Puzzle problem



Classic AI Search Problems

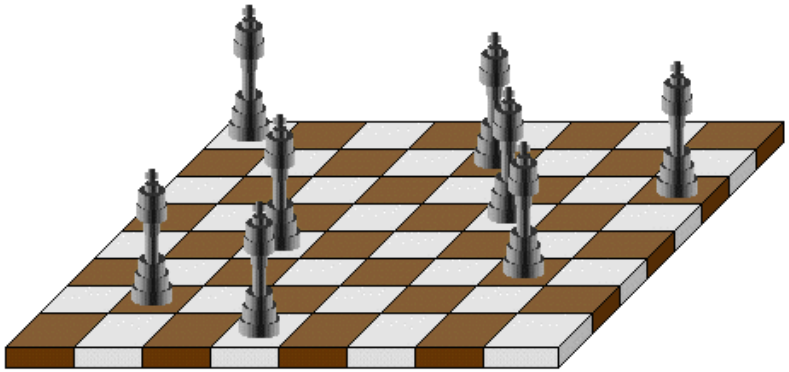
- N-Queens:
- Problem of placing n chess queens on an n×n chessboard so that no two queens attack each other
- A solution requires that no two queens share the same row, column, or diagonal

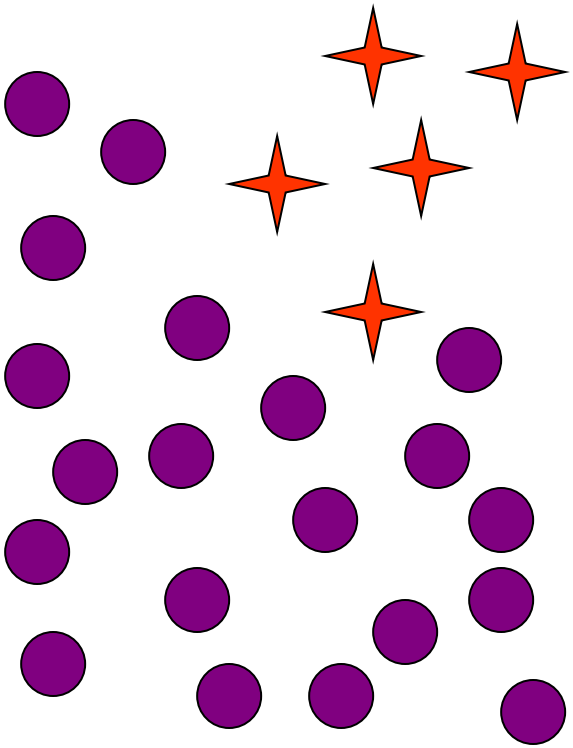


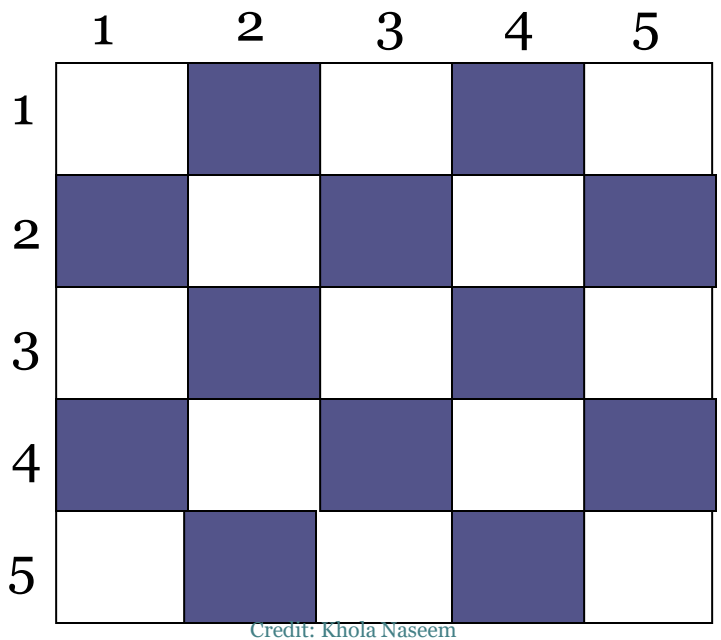
Credit: Khola Naseem

Classic AI Search Problems

- 5-Queens:



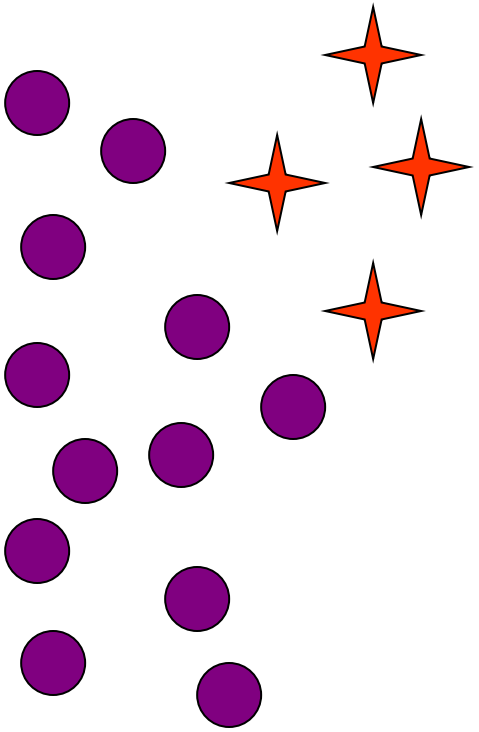


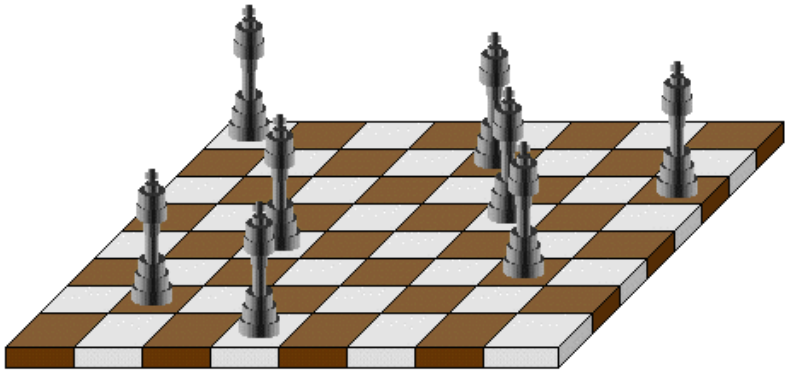


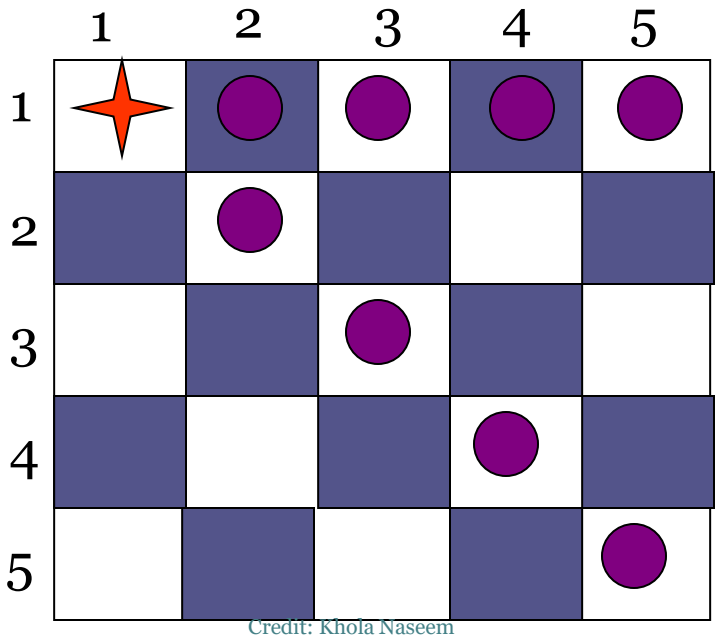
Credit: Khola Naseem

Classic AI Search Problems

- 5-Queens:



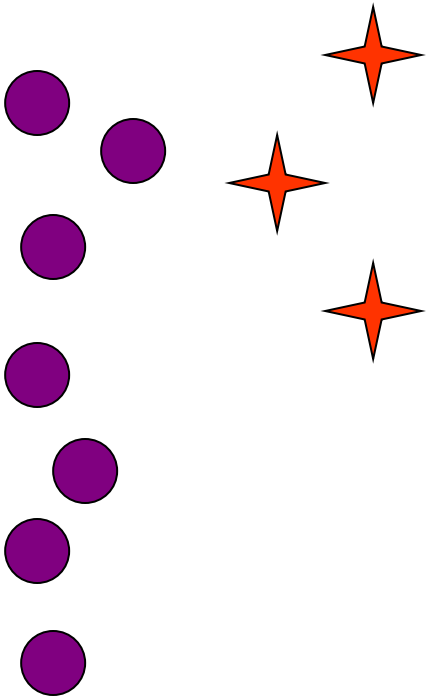


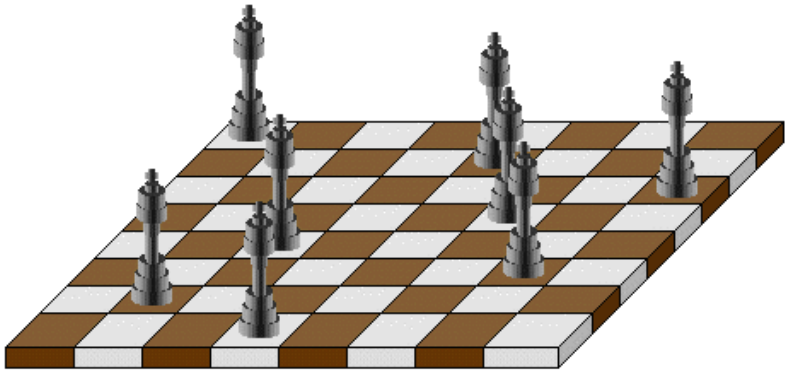


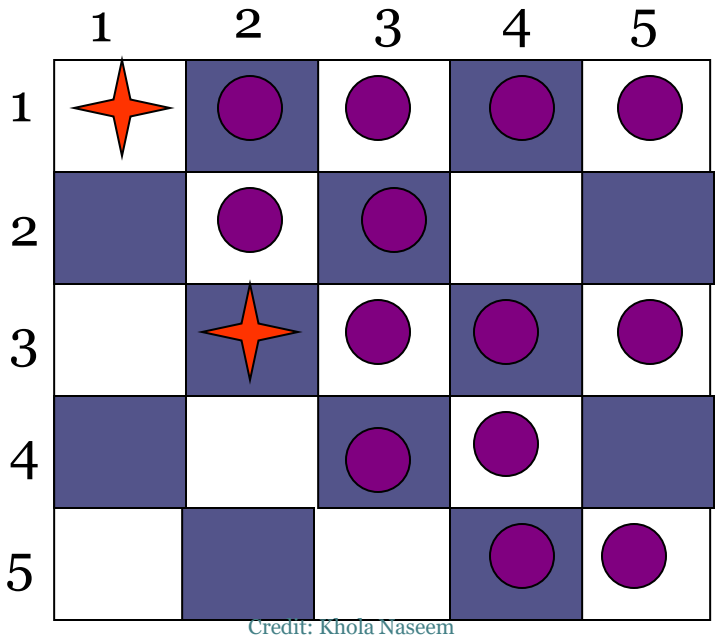
Credit: Khola Naseem

Classic AI Search Problems

- 5-Queens:



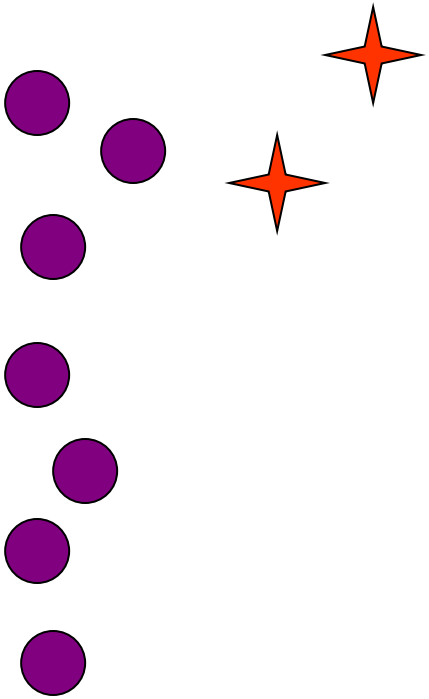


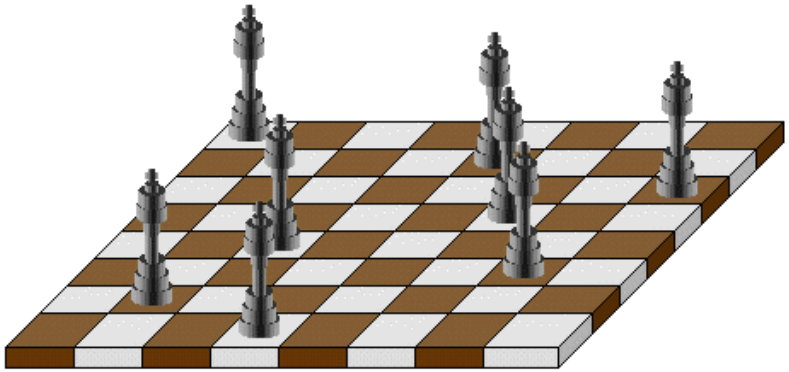


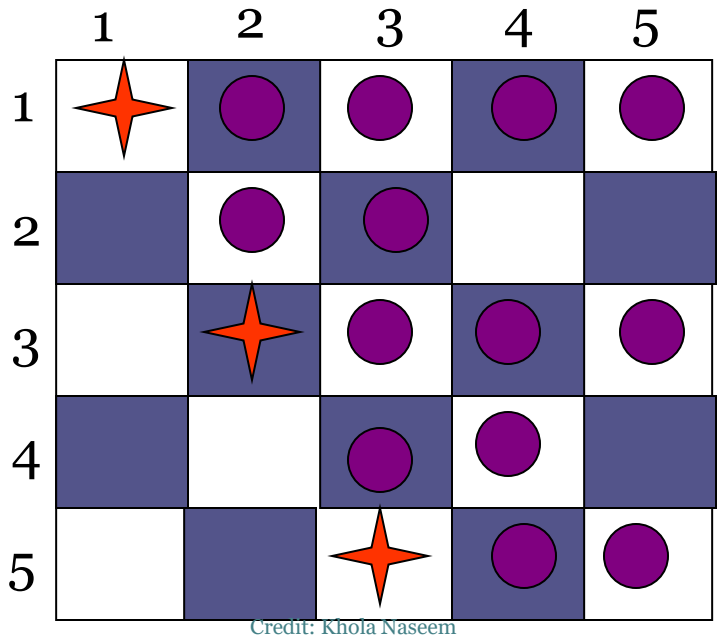
Credit: Khola Naseem

Classic AI Search Problems

- 5-Queens:



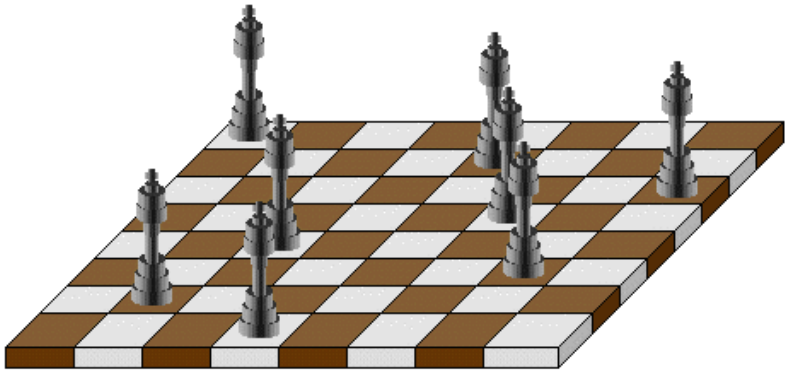


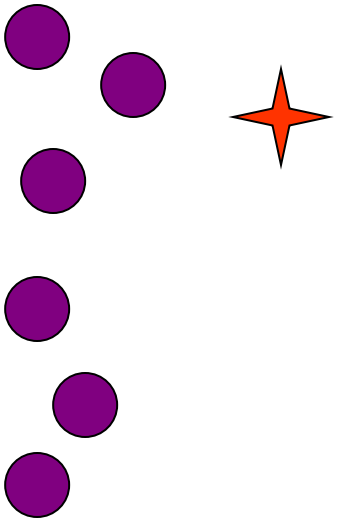


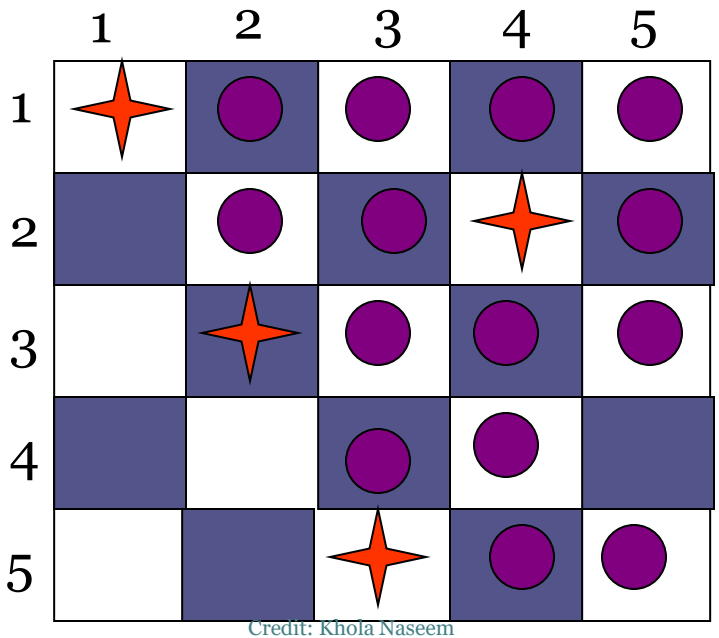
Credit: Khola Naseem

Classic AI Search Problems

- 5-Queens:



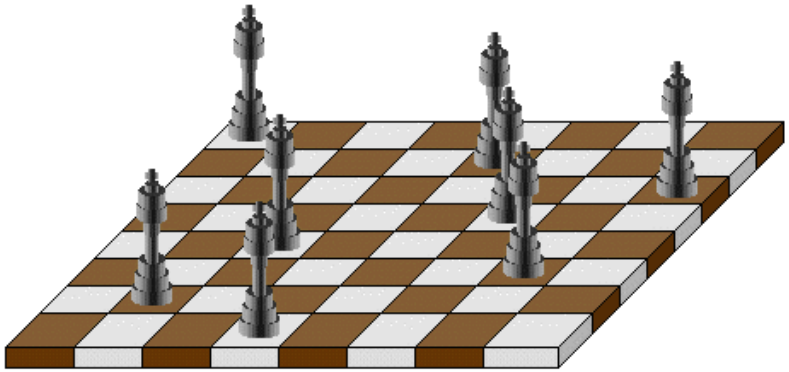


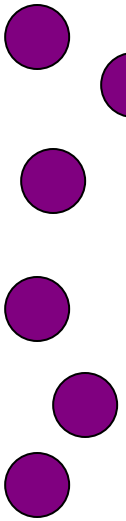


Credit: Khola Naseem

Classic AI Search Problems

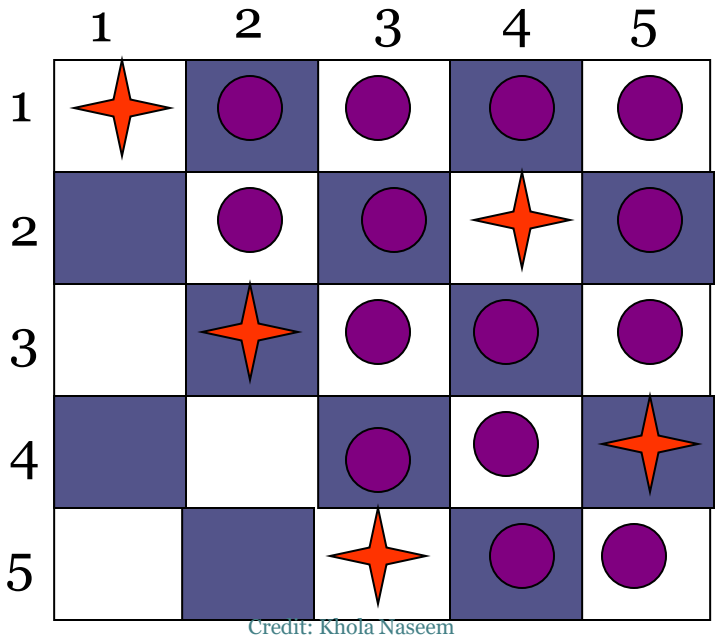
- 5-Queens:





Solution !!

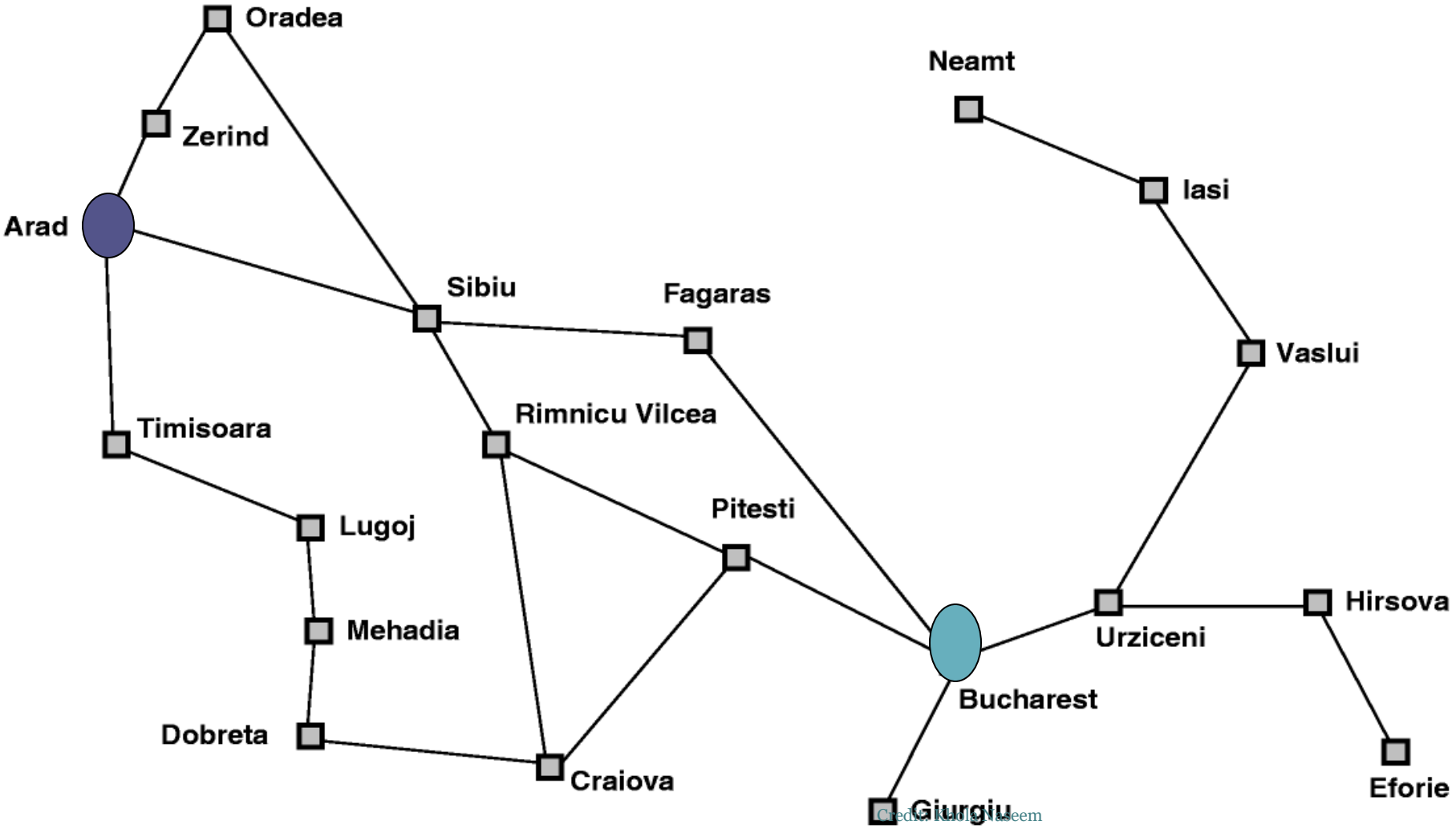
No Queen is under Attack



Credit: Khola Naseem

Classic AI Search Problems

- Map searching (navigation)



Problem Solving by Searching

- Problem-solving agent:
  - Is a kind of goal-based agent
- It solves problem by
  - finding sequences of actions that lead to desirable states (goals)
- To solve a problem,
  - the first step is the goal formulation, based on the current situation

Problem Solving by Searching

- The goal is formulated
  - as a set of world states, in which the goal is satisfied
- Reaching from initial state to goal state
- Actions are required
  - Actions are the operators
  - causing transitions between world states

Problem Solving by Searching

The Problem formulation

- The process of deciding
  - what actions and states to consider
- E.g., driving Main campus to Ksk
- in-between states and actions defined
- States: Some places between main campus & Ksk
- Actions: Turn left, Turn right, go straight, accelerate & brake, etc

Credit: Khola Naseem

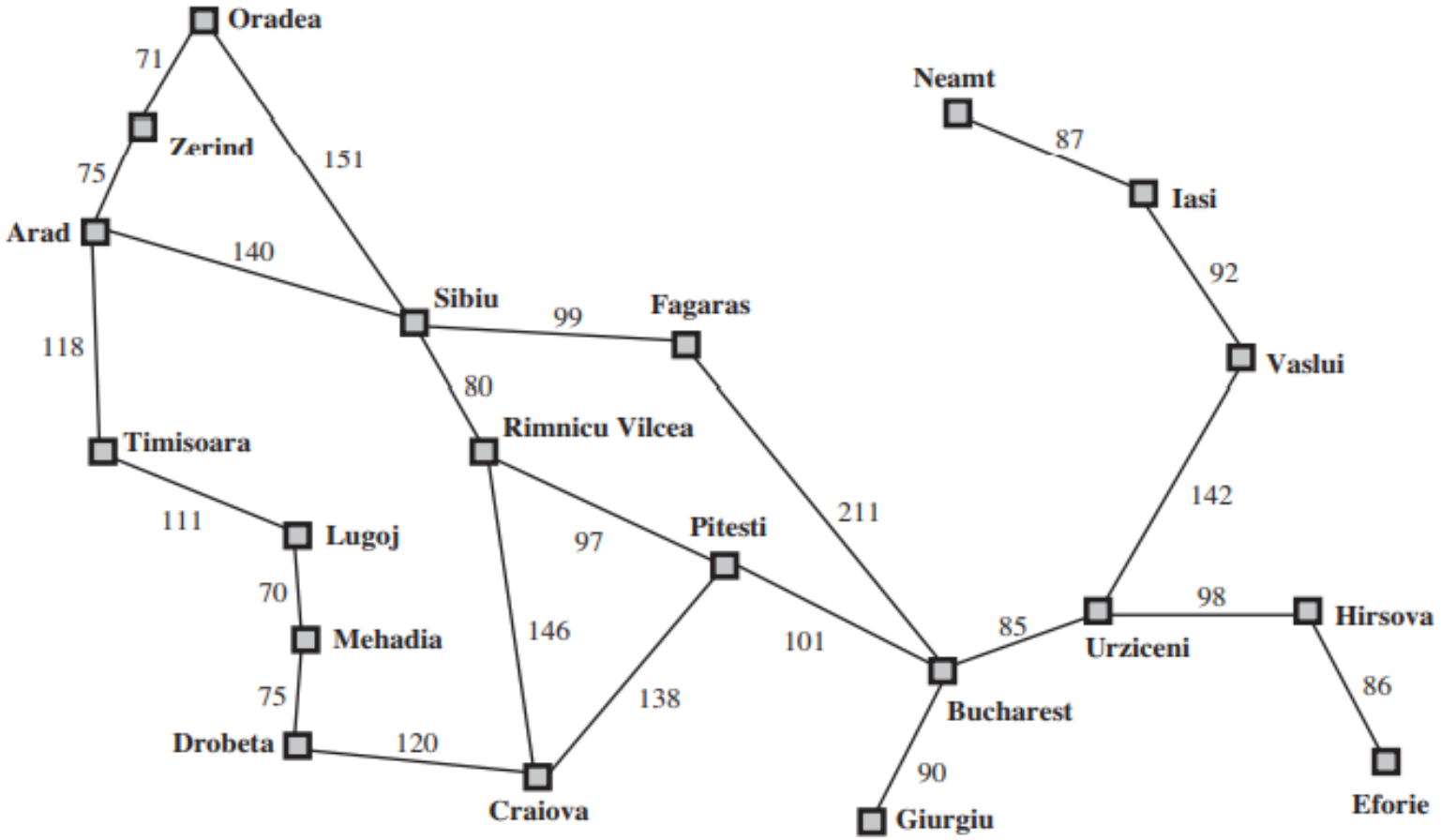
Problem Solving by Searching

Why searching is important:

- Because there are many ways to achieve the same goal
- the agent can examine different possible sequences of actions, and choose the best
- The process of looking for a sequence of actions that reaches the goal is called search
- The best sequence is then a list of actions, called solution
- Once a solution is found, the actions it recommends can be carried out. This EXECUTION is called the execution phase.

Problem Solving by Searching

- Example: Romania map
- Goal: Go to Bucharest from Arad



Credit: Khola Naseem

Well-defined problems and solutions:

- A problem can be defined formally by five components:
  - Initial state
  - Actions
  - Transition model or (Successor functions)
  - Goal Test
  - Path Cost

Well-defined problems and solutions:

- Initial state:
  - The initial state that the agent starts in. For example, the initial state for our agent in Romania might be described as In(Arad).
- Action:
  - A description of the possible actions available to the agent. Given a particular state s, ACTIONS(s) returns the set of actions that can be executed in s. We say that each of these actions is applicable in s.
  - For example, from the state In(Arad), the applicable actions are {Go(Sibiu), Go(Timisoara), Go(Zerind)}.

Well-defined problems and solutions:

- TRANSITION MODEL :
  - A description of what each action does; the formal name for this is the transition. TRANSITION MODEL model, specified by a function RESULT(s, a) that returns the state that results from SUCCESSOR doing action a in state s.
  - We also use the term successor to refer to any state reachable from a given state by a single action.
  - For example, we have RESULT(In(Arad),Go(Zerind)) = In(Zerind) .

Well-defined problems and solutions:

- State space :
  - Together, the initial state, actions, and transition model implicitly define the **state space** of the problem
  - the set of all states reachable from the initial state by any sequence of actions.
- Path:
  - A path in the state space is a sequence of states connected by a sequence of actions.

Well-defined problems and solutions:

Goal test

- The goal test, which determines whether a given state is a goal state.
- Sometimes there is an explicit set of possible goal states, and the test simply checks whether the given state is one of them.
- The agent's goal in Romania is the singleton set {In(Bucharest)}

Well-defined problems and solutions:

path cost:

- Is a function
- assigns a numeric cost to each path
- = performance measure
- denoted by g
- to distinguish the best path from others
- Usually the path cost is the sum of the step costs of the individual actions

Well-defined problems and solutions:

- Solution:
  - A solution to a problem is an action sequence that leads from the initial state to a goal state.
- OPTIMAL SOLUTION
  - Solution quality is measured by the OPTIMAL SOLUTION path cost function, and an optimal solution has the lowest path cost among all solutions

Well-defined problems and solutions:

On holiday in Romania; currently in Arad. Flight leaves tomorrow from Bucharest

- Formulate goal:
  - be in Bucharest
- Formulate problem:
  - states: various cities
  - actions: drive between cities
- Find solution:
  - sequence of cities, e.g., Arad, Sibiu, Fagaras, Bucharest

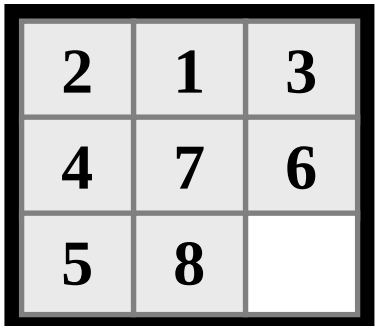
Credit: Khola Naseem

Well-defined problems and solutions:

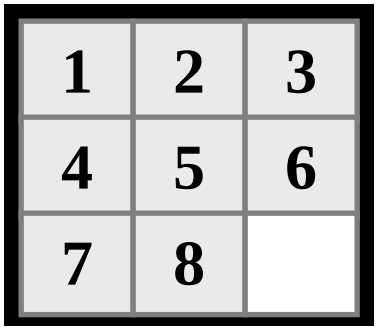
- A simple problem-solving agent first formulates a goal and a problem, searches for a sequence of actions that would solve the problem, and then executes the actions one at a time. When this is complete, it formulates another goal and starts over.

Problem Formulation
8-Puzzle Problem

Initial State	Operators	Goal State



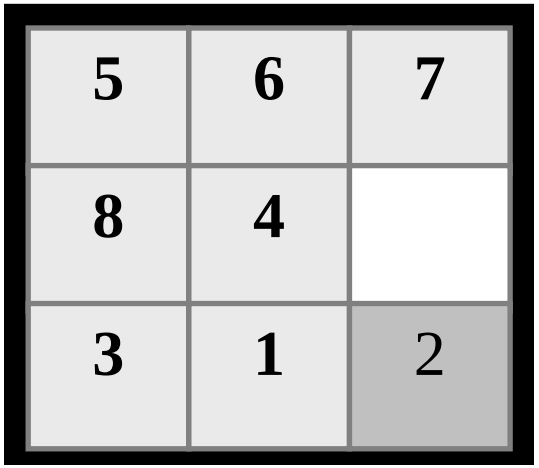
Slide blank square left.
Slide blank square right.
….



Credit: Khola Naseem

Problem Formulation 8-Puzzle Problem

- **Representing states:**
- **For the 8-puzzle**
- 3 by 3 array
  - 5, 6, 7
  - 8, 4, BLANK
  - 3, 1, 2
- A vector of length nine
  - 5, 6, 7, 8, 4, BLANK, 3, 1, 2
- A list of facts
  - Upper_left = 5
  - Upper_middle = 6
  - Upper_right = 7
  - Middle_left = 8



Credit: Khola Naseem

Problem Formulation 8-Puzzle Problem

Specifying operators

There are often many ways to specify the operators, some will be much easier to implement…

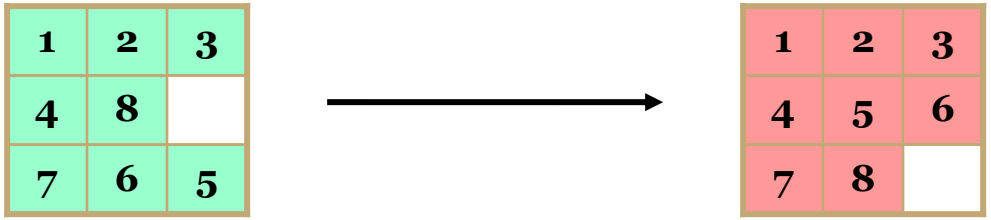
- **Move 1 left**
- **Move 1 right**
- **Move 1 up**
- **Move 1 down**
- **Move 2 left**
- **Move 2 right**
- **Move 2 up**
- **Move 2 down**
- **Move 3 left**
- **Move 3 right**
- **Move 3 up**
- **Move 3 down**
- **Move 4 left**
- …

- **Move Blank left**
- **Move Blank right**
- **Move Blank up**
- **Move Blank down**

5	6	7
8	4	
3	1	2

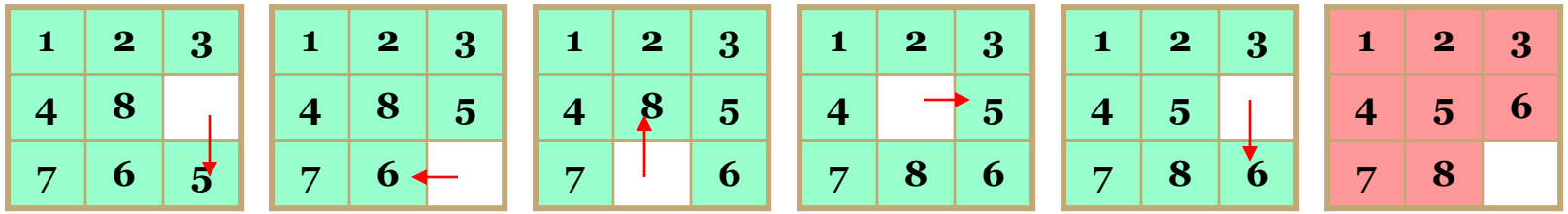
Credit: Khola Naseem

Problem Formulation 8-Puzzle Problem



Initial state **Goal state**

Operators: ***slide blank up, slide blank down, slide blank left, slide blank right***



Solution: **sb-down, sb-left, sb-up,sb-right, sb-down**

Path cost: **5** steps to reach the goal

Credit: Khola Naseem

Problem Formulation 8-Puzzle Problem

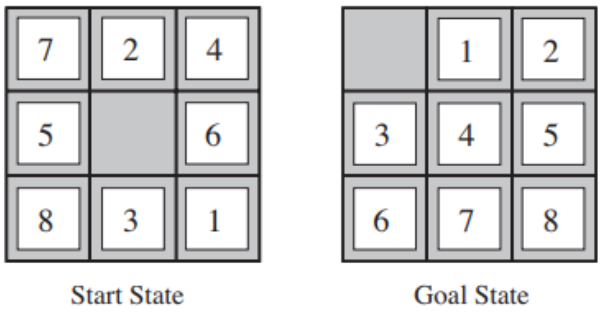


Figure 3.4 A typical instance of the 8-puzzle.

- **States**: A state description specifies the location of each of the eight tiles and the blank in one of the nine squares.
- **Initial state**: Any state can be designated as the initial state.

- **Actions**: The simplest formulation defines the actions as movements of the blank space *Left*, *Right*, *Up*, or *Down*. Different subsets of these are possible depending on where the blank is.
- **Transition model**: Given a state and action, this returns the resulting state; for example, if we apply *Left* to the start state in Figure 3.4, the resulting state has the 5 and the blank switched.
- **Goal test**: This checks whether the state matches the goal configuration shown in Figure 3.4. (Other goal configurations are possible.)
- **Path cost**: Each step costs 1, so the path cost is the number of steps in the path.

Credit: Khola Naseem

Missionaries and cannibals

- Three missionaries and three cannibals are on the left bank of a river.
- There is one boat which can hold one or two people.
- Find a way to get everyone to the right bank, without ever leaving a group of missionaries in one place outnumbered by cannibals in that place.

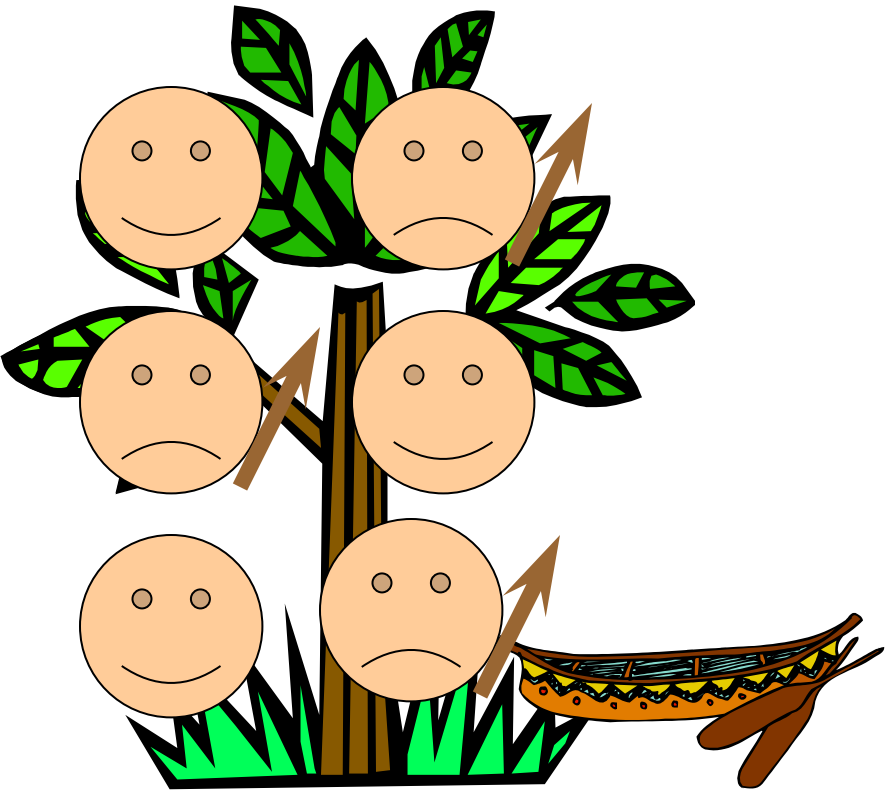
Credit: Khola Naseem

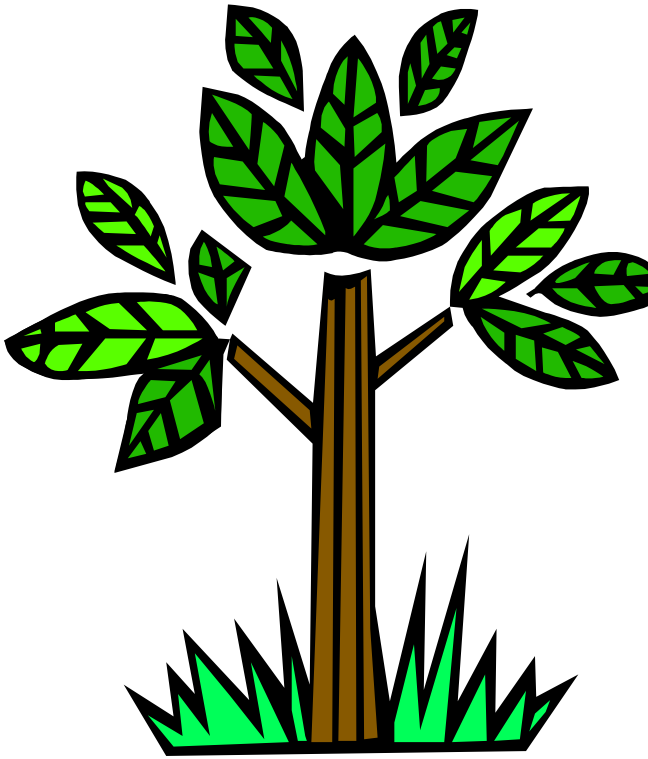
Missionaries and cannibals

- States: three numbers (i, j, k) representing the number of
  - missionaries,
  - cannibals,
  - boats on the left bank of the river.
- Initial state: (3, 3, 1)
- **Operators:** in a given direction, take
  - one missionary,
  - one cannibal,
  - two missionaries,
  - two cannibals,
  - one missionary and one cannibal across the river
- Goal Test: reached state (0, 0, 0)?
- Path Cost: Number of crossings.

Missionaries and Cannibals

(3,3,1): Initial State

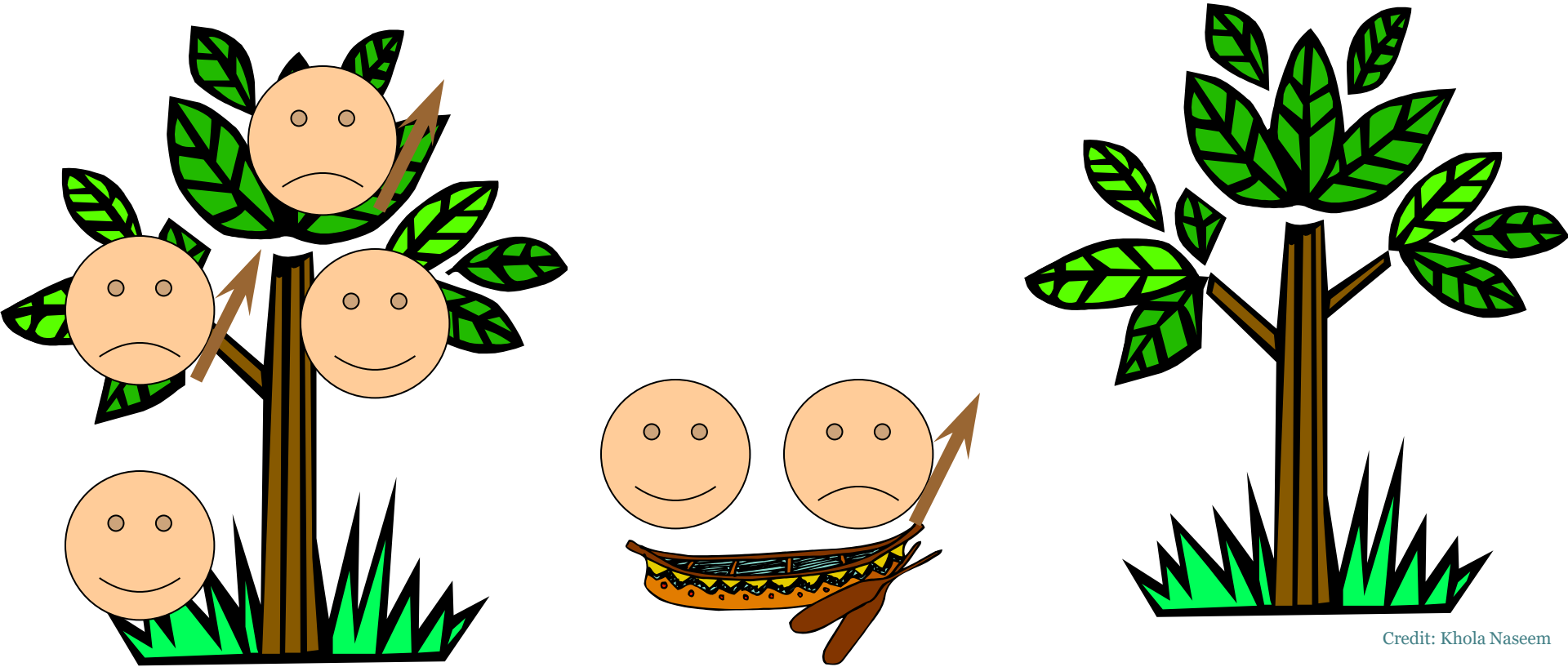




Credit: Khola Naseem

Missionaries and Cannibals

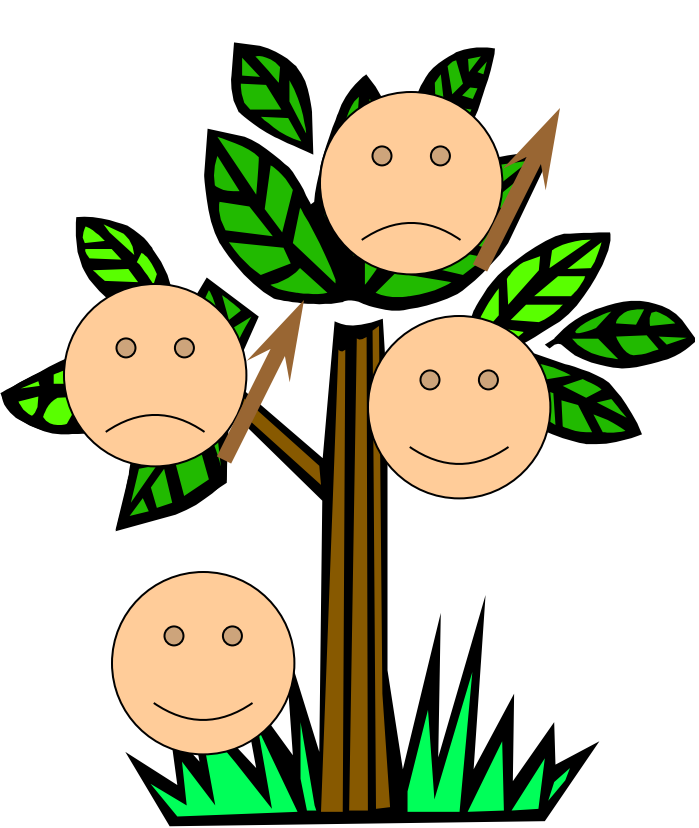
A missionary and cannibal cross



Credit: Khola Naseem

Missionaries and Cannibals

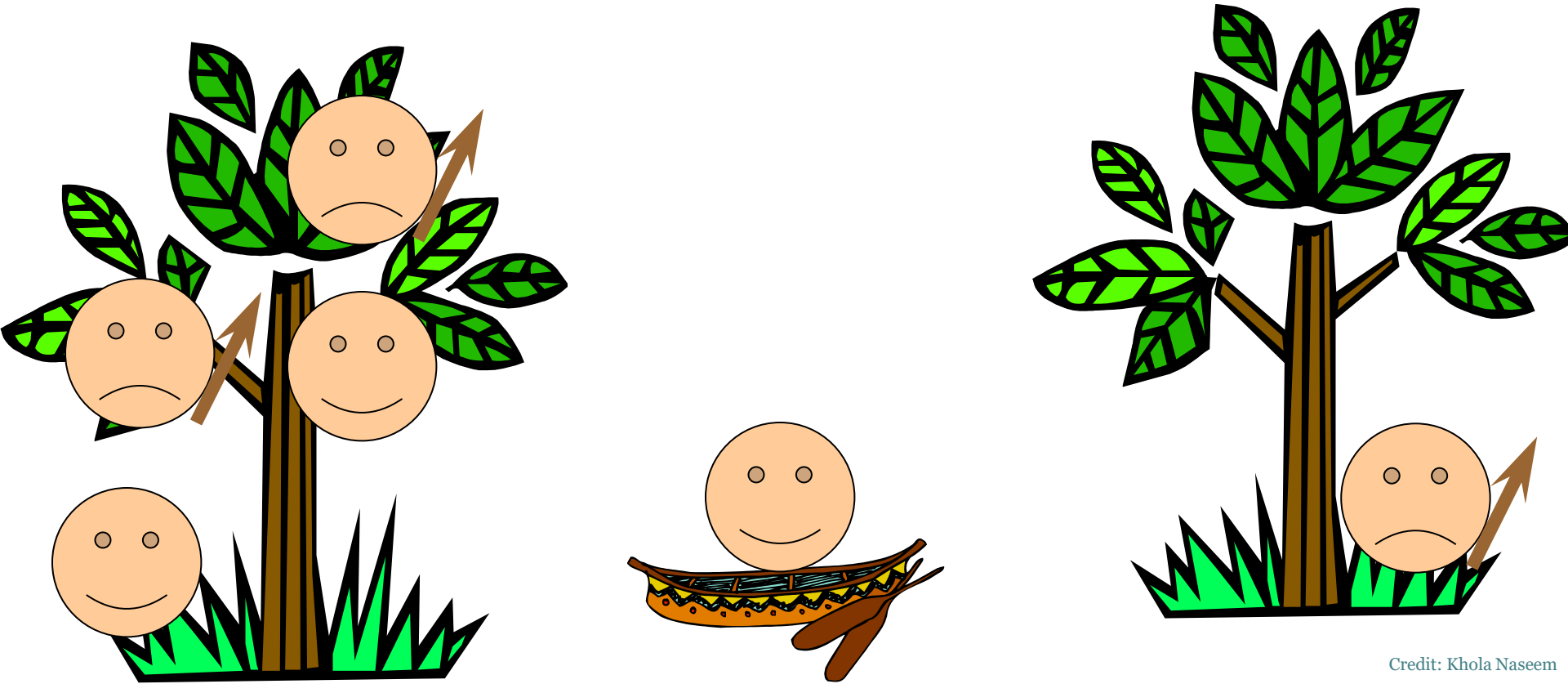
(2,2,0)



Credit: Khola Naseem

Missionaries and Cannibals

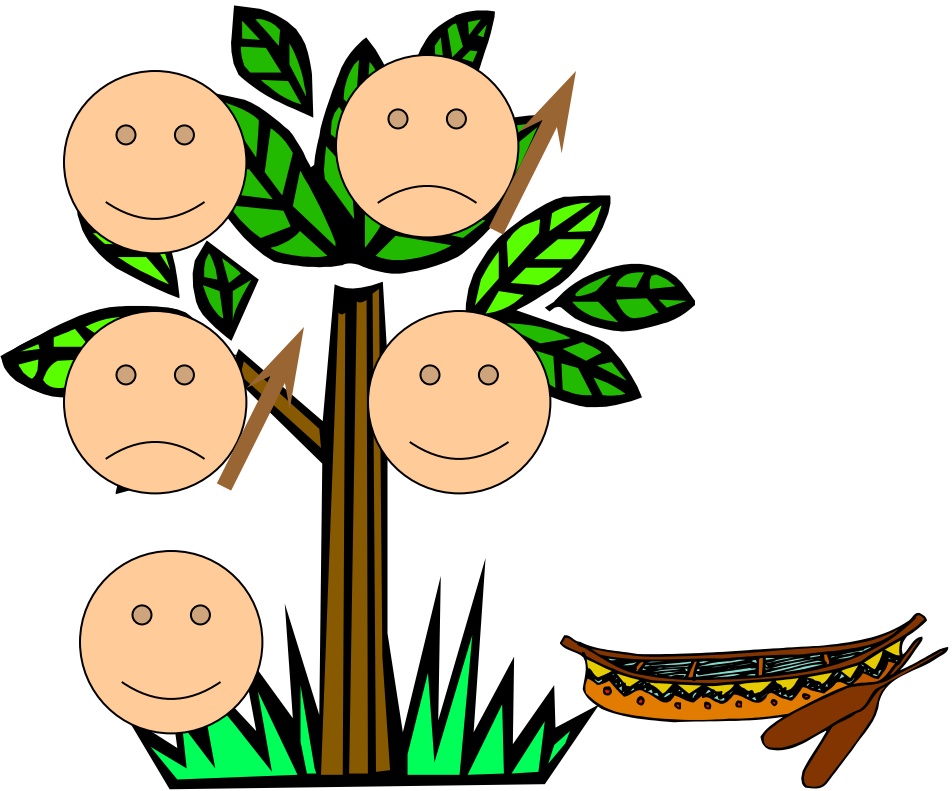
One missionary returns

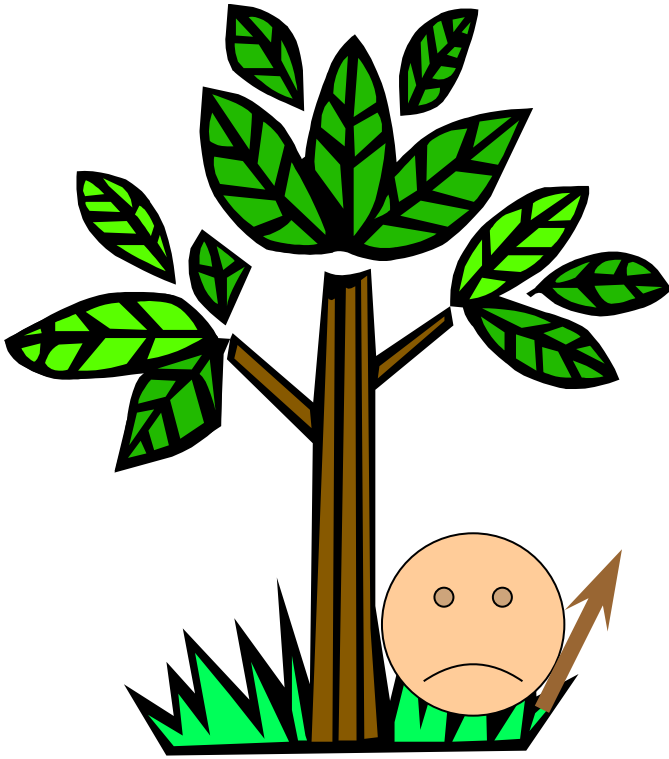


Credit: Khola Naseem

Missionaries and Cannibals

(3,2,1)

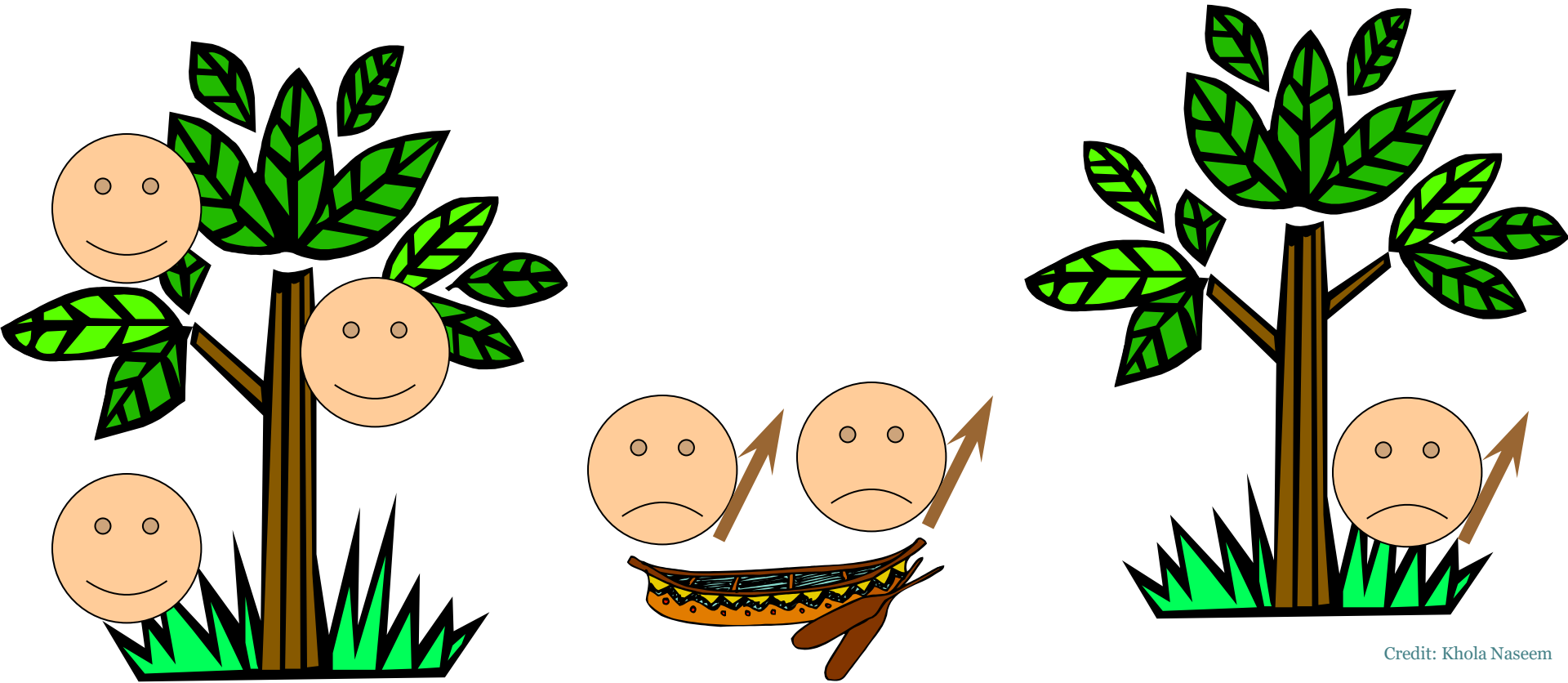




Credit: Khola Naseem

Missionaries and Cannibals

Two cannibals cross



Credit: Khola Naseem

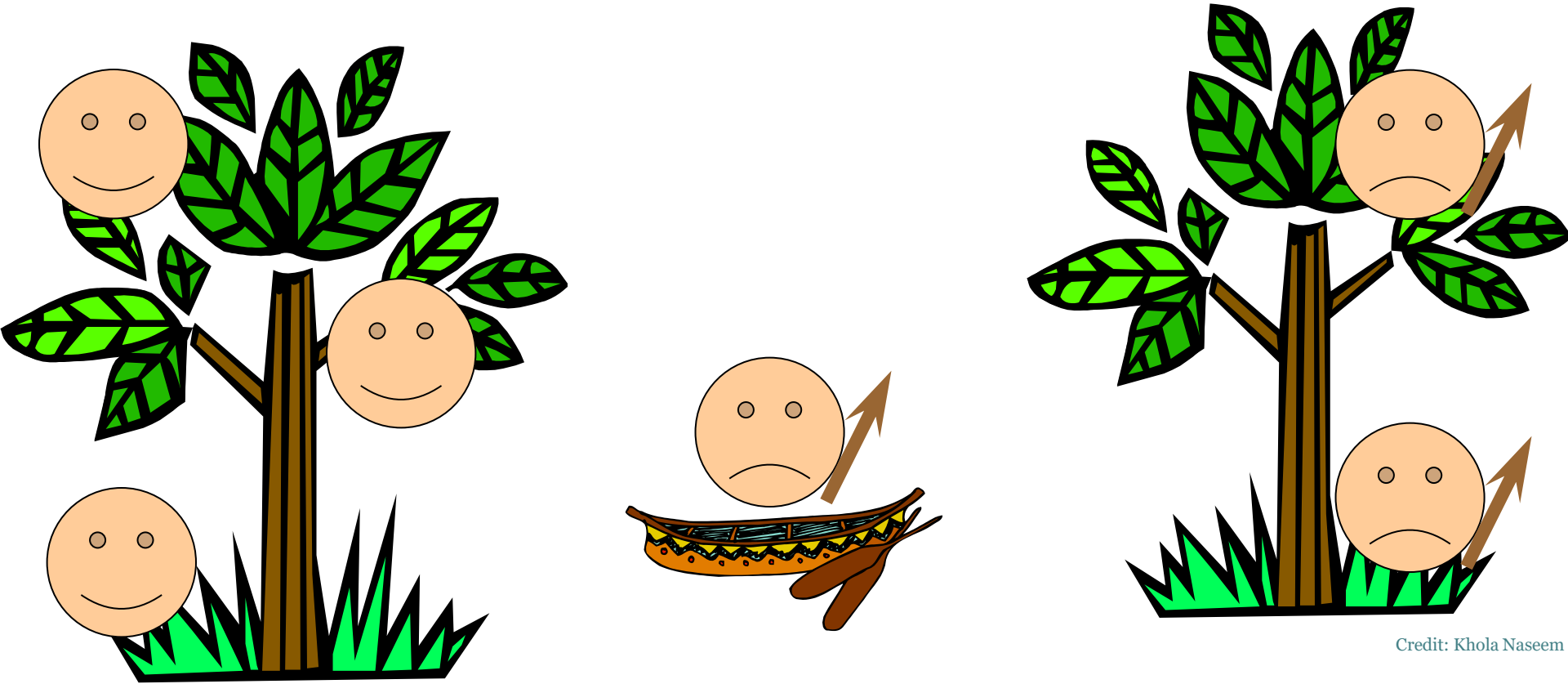
Missionaries and Cannibals

(3,0,0)

Credit: Khola Naseem

Missionaries and Cannibals

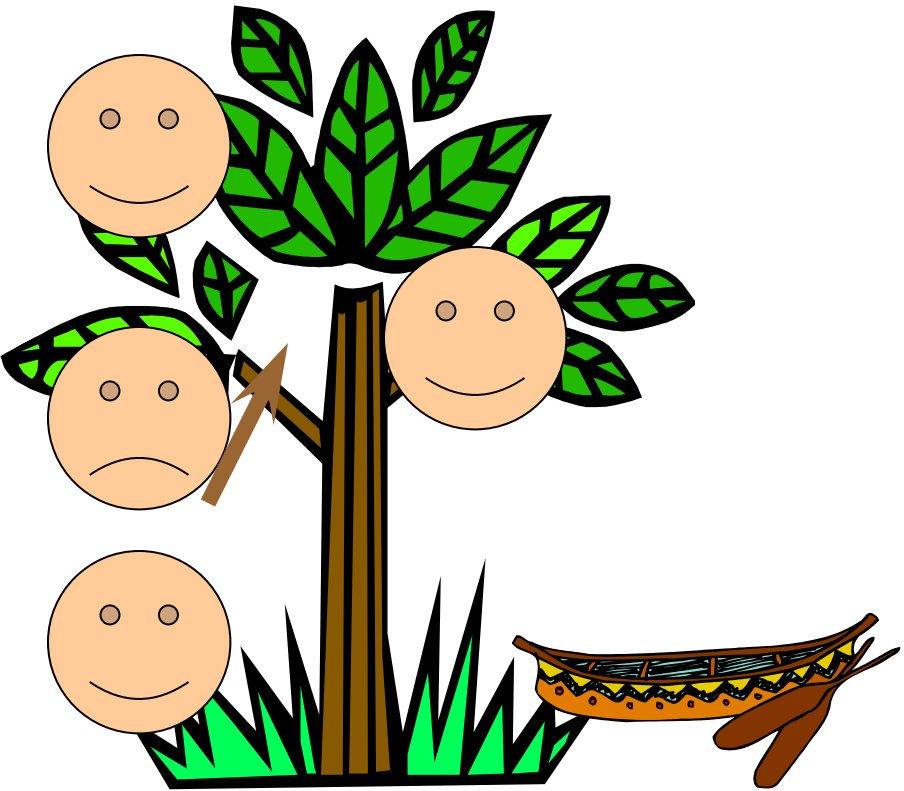
A cannibal returns

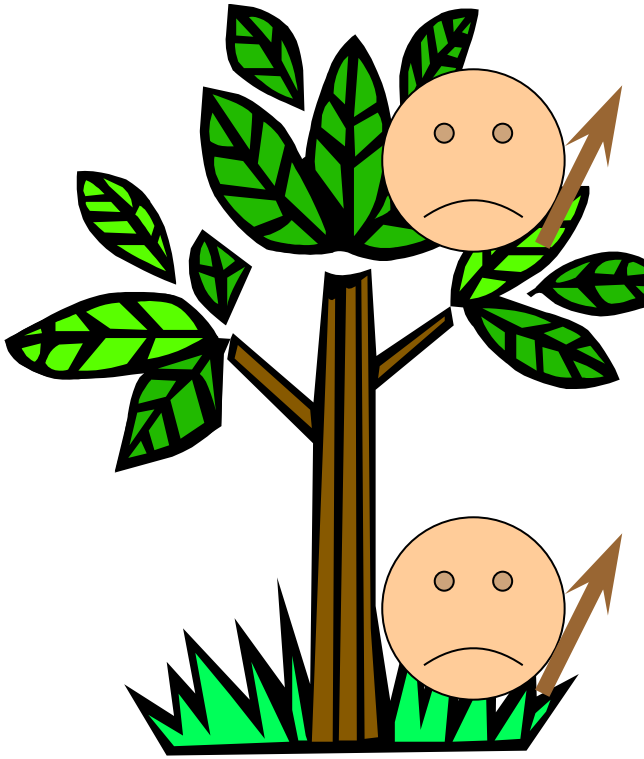


Credit: Khola Naseem

Missionaries and Cannibals

(3,1,1)

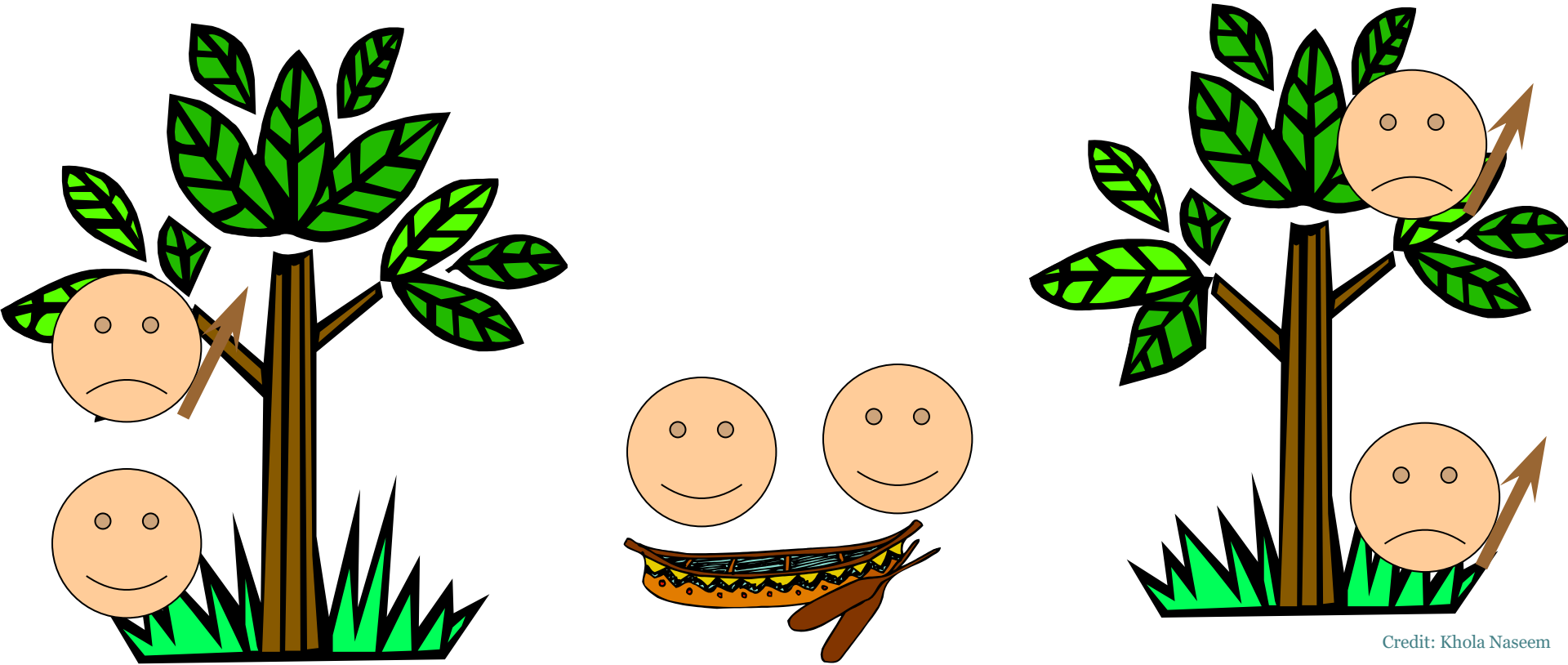




Credit: Khola Naseem

Missionaries and Cannibals

Two missionaries cross



Credit: Khola Naseem

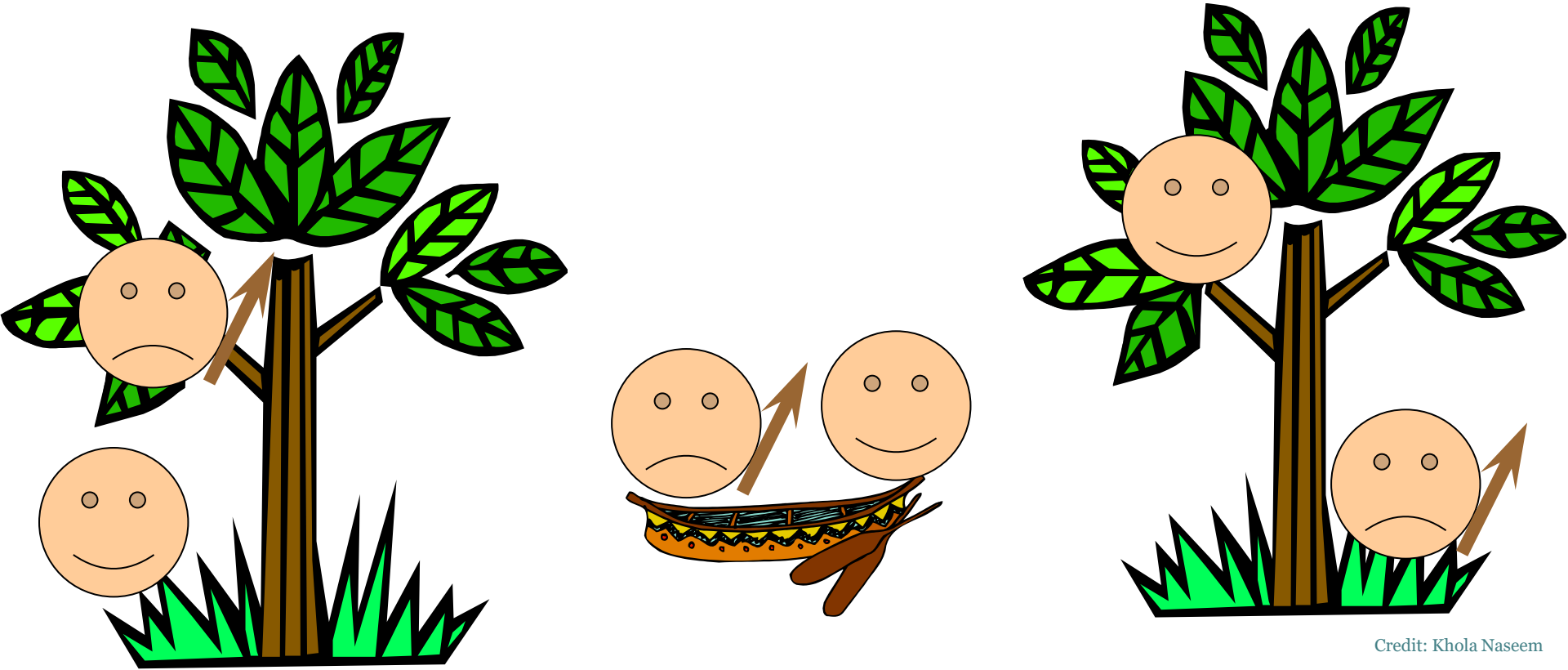
Missionaries and Cannibals

(1,1,0)

Credit: Khola Naseem

Missionaries and Cannibals

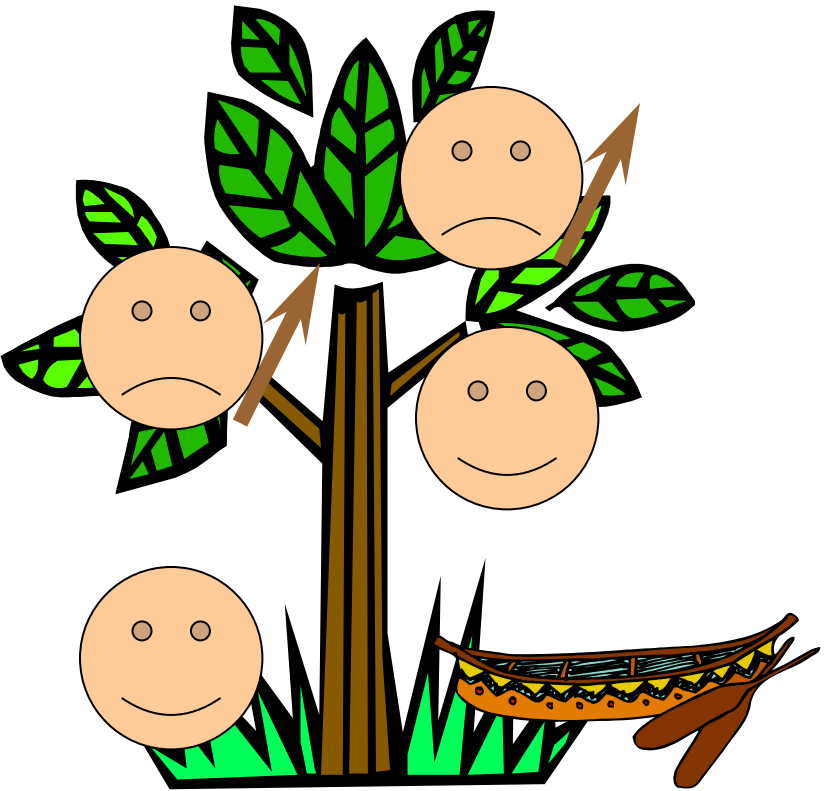
A missionary and cannibal return

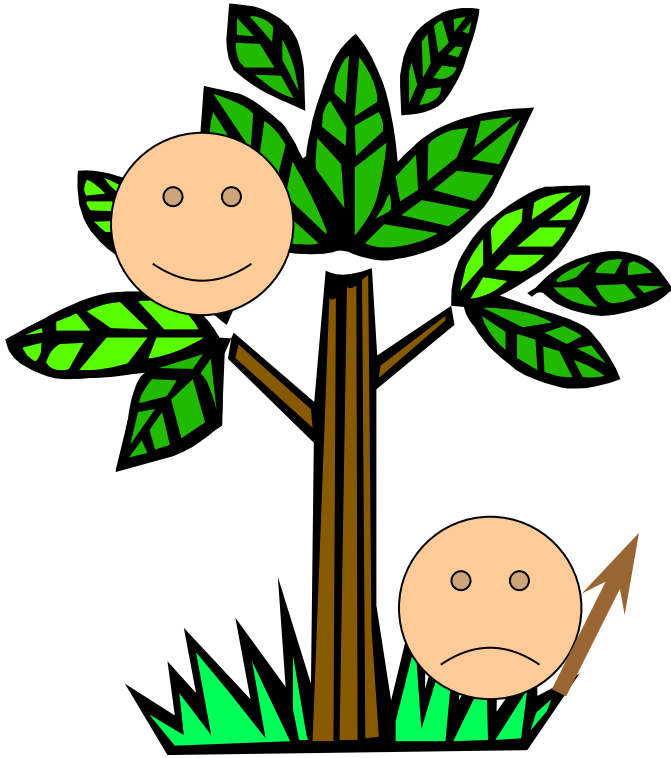


Credit: Khola Naseem

Missionaries and Cannibals

(2,2,1)

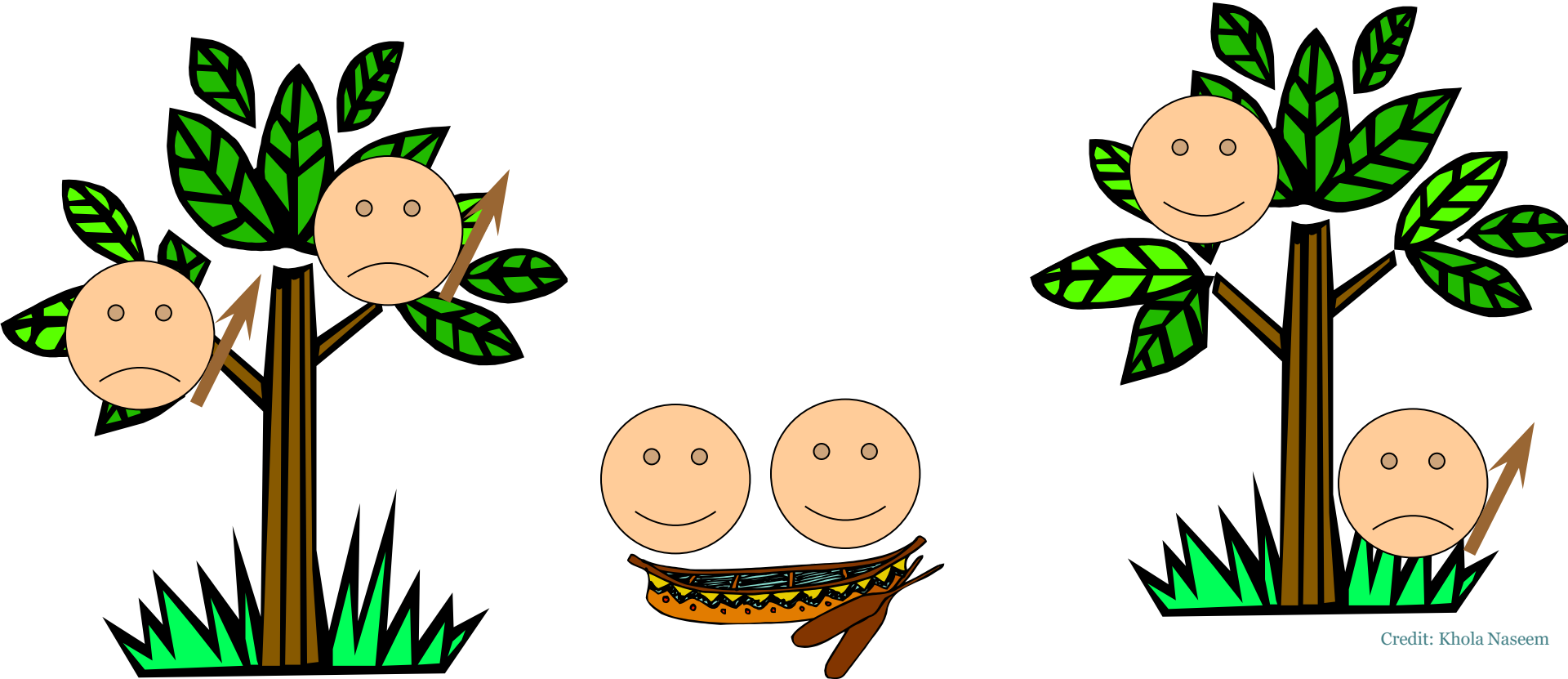




Credit: Khola Naseem

Missionaries and Cannibals

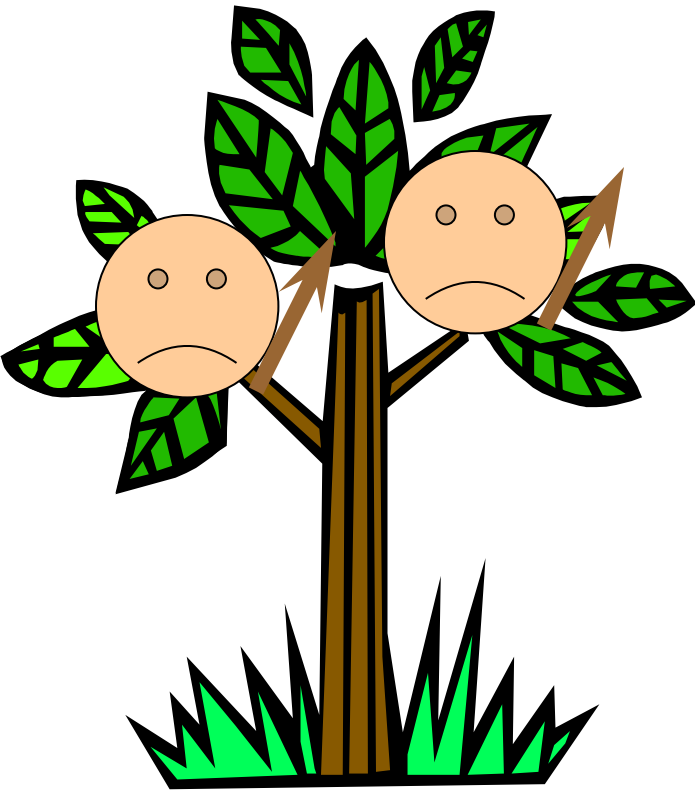
Two Missionaries cross

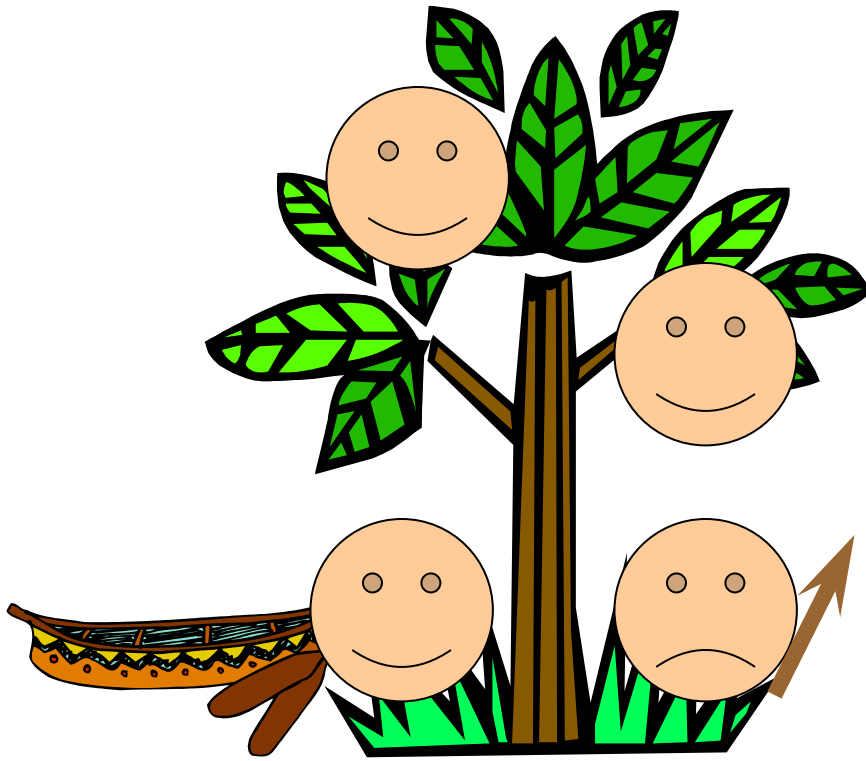


Credit: Khola Naseem

Missionaries and Cannibals

(0,2,0)

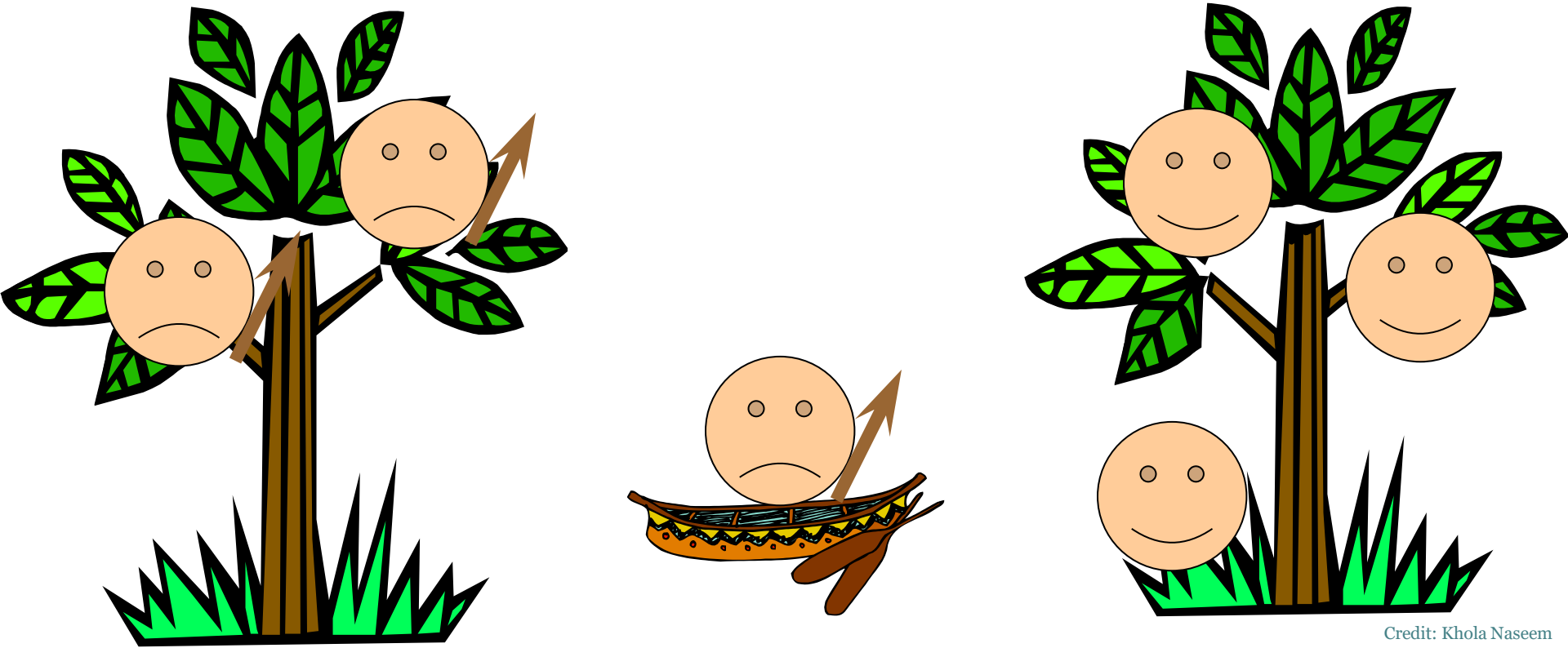




Credit: Khola Naseem

Missionaries and Cannibals

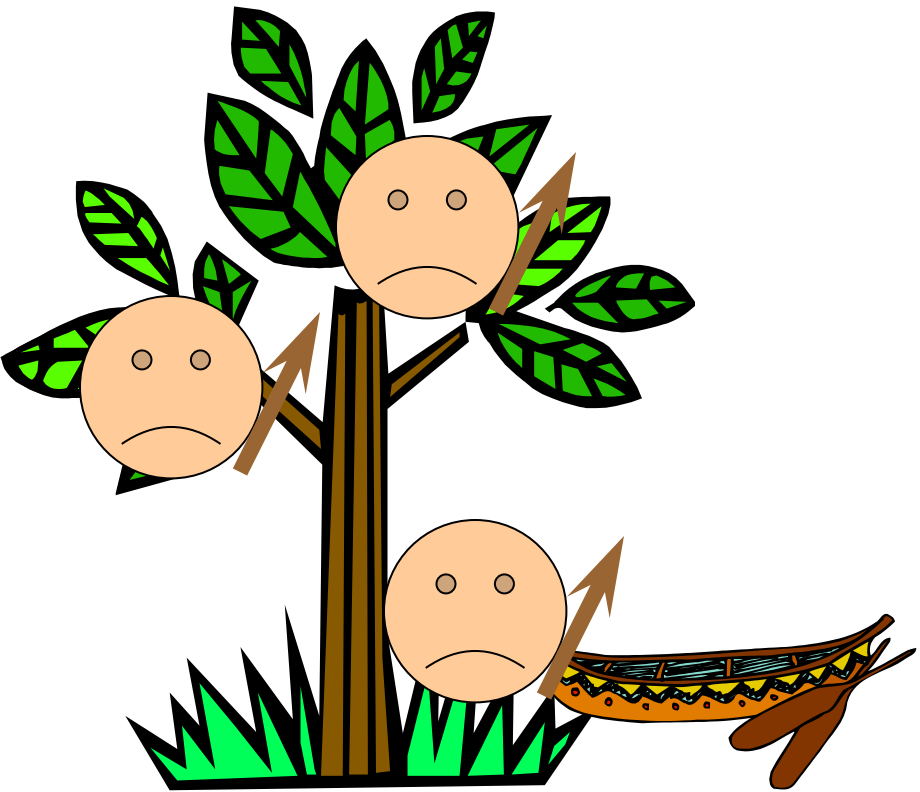
A cannibal returns

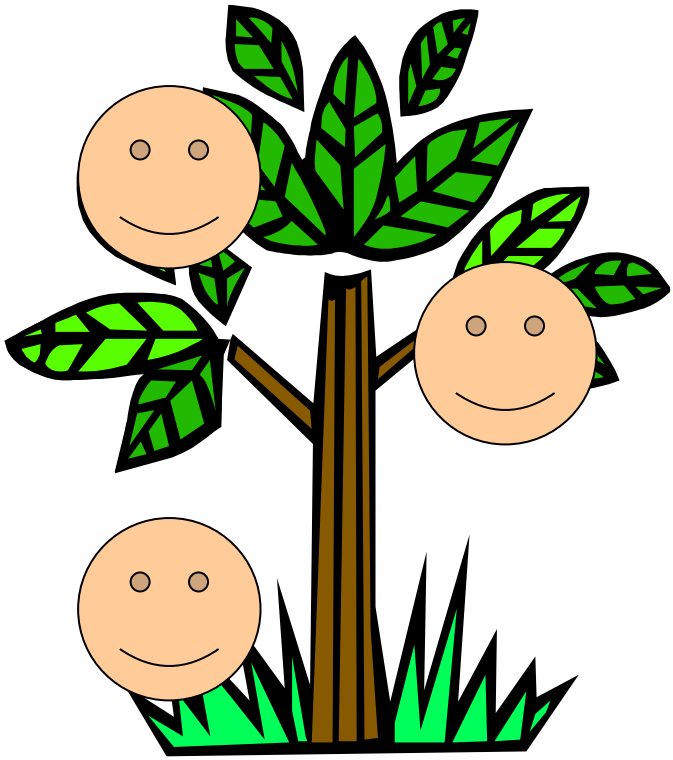


Credit: Khola Naseem

Missionaries and Cannibals

(0,3,1)





Credit: Khola Naseem

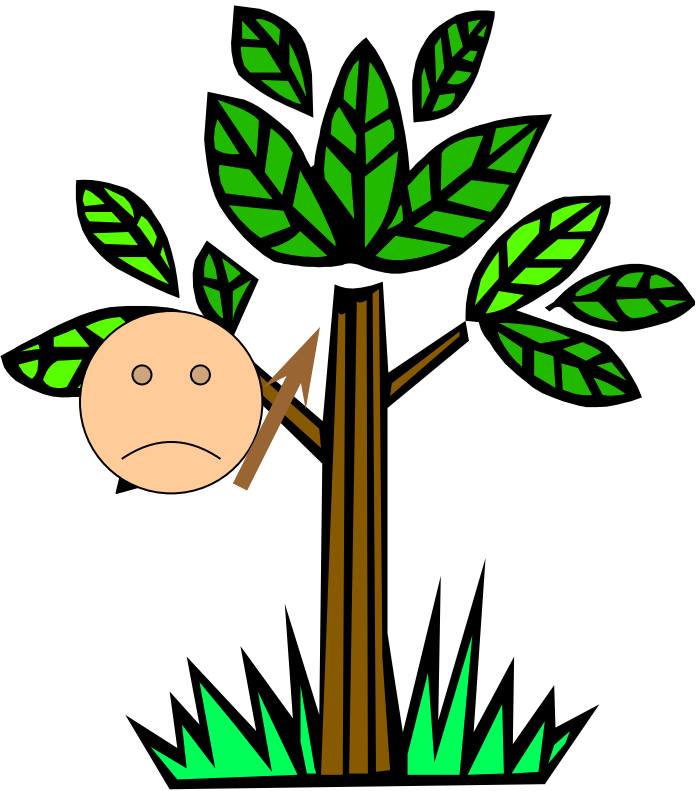
Missionaries and Cannibals

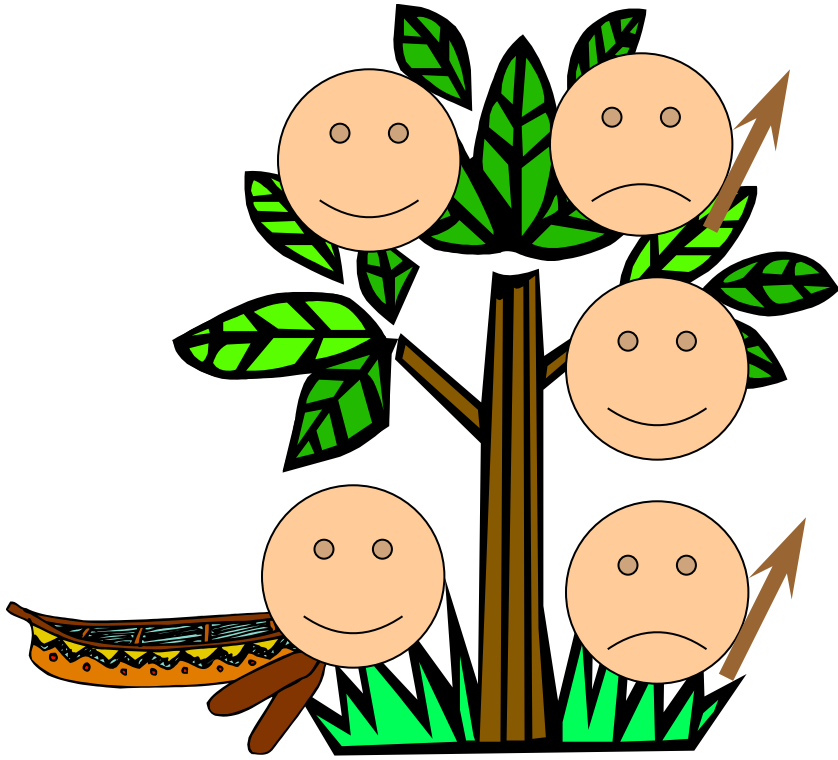
Two cannibals cross

Credit: Khola Naseem

Missionaries and Cannibals

(0,1,0)

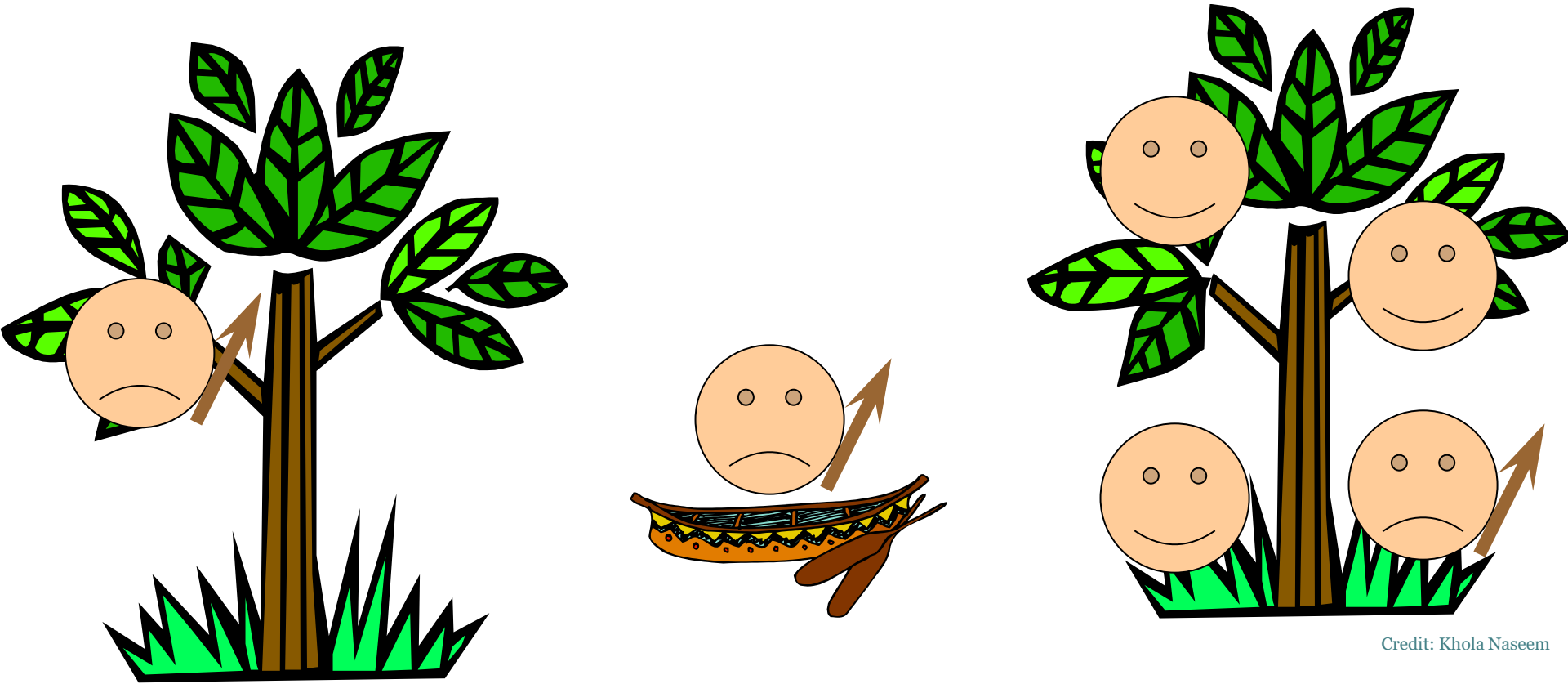




Credit: Khola Naseem

Missionaries and Cannibals

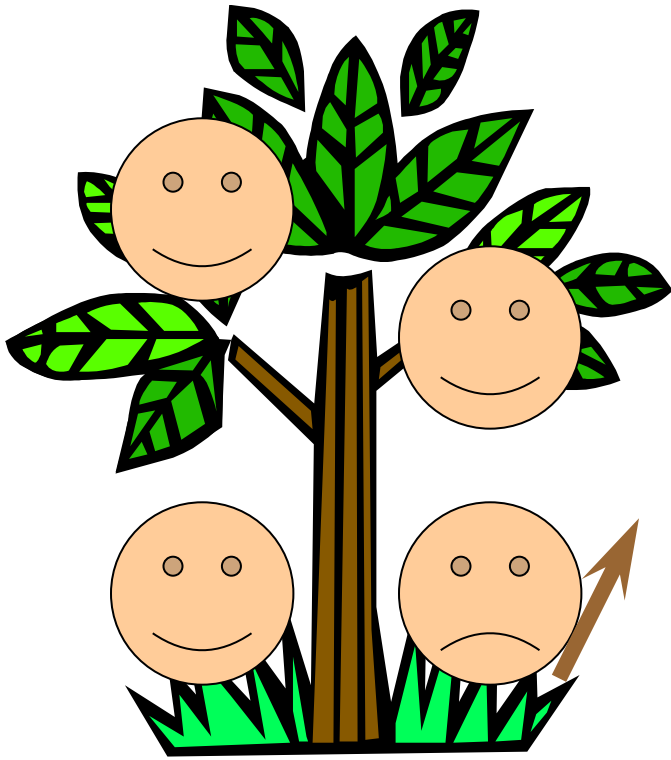
A cannibal returns



Credit: Khola Naseem

Missionaries and Cannibals

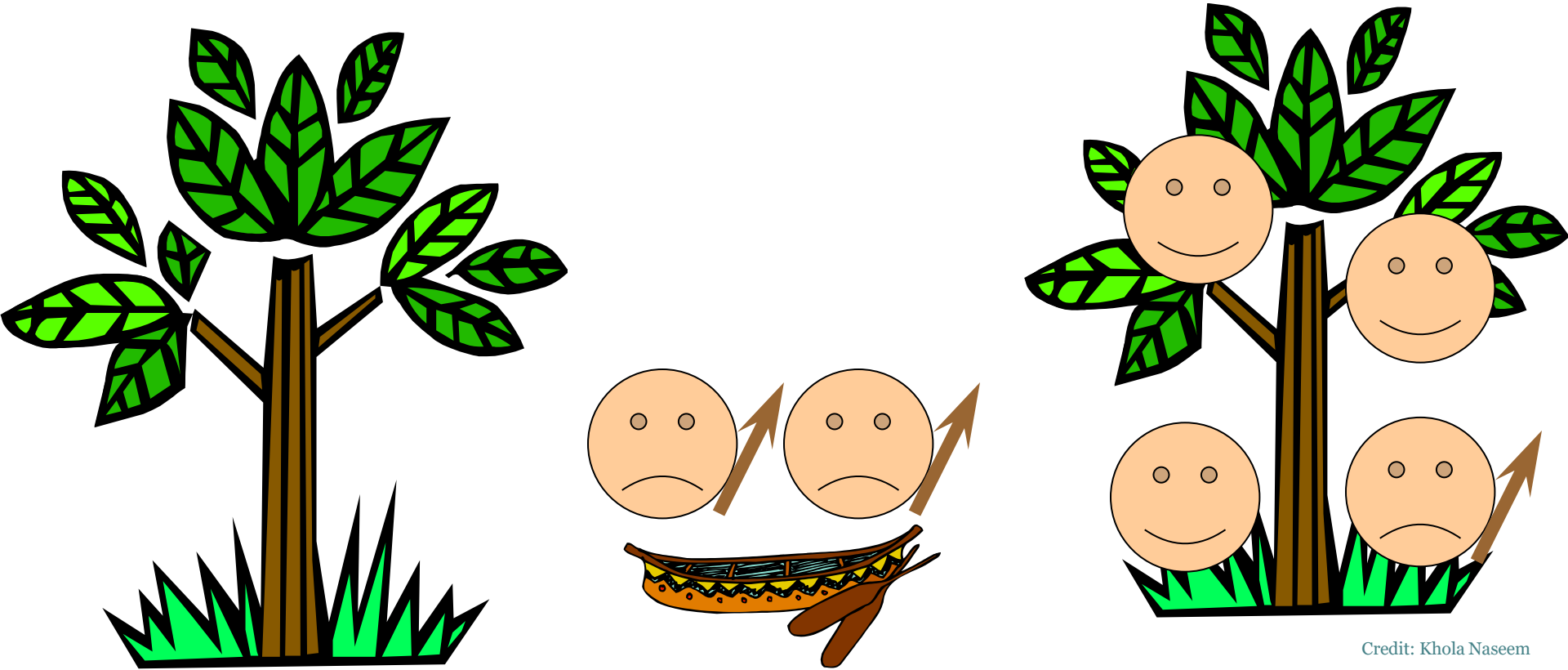
(0,2,1)



Credit: Khola Naseem

Missionaries and Cannibals

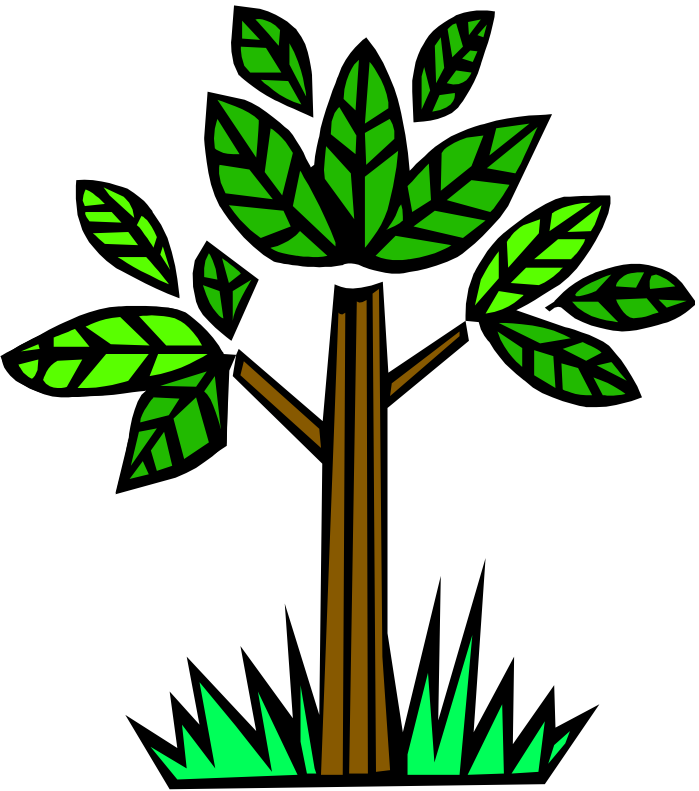
The last two cannibals cross

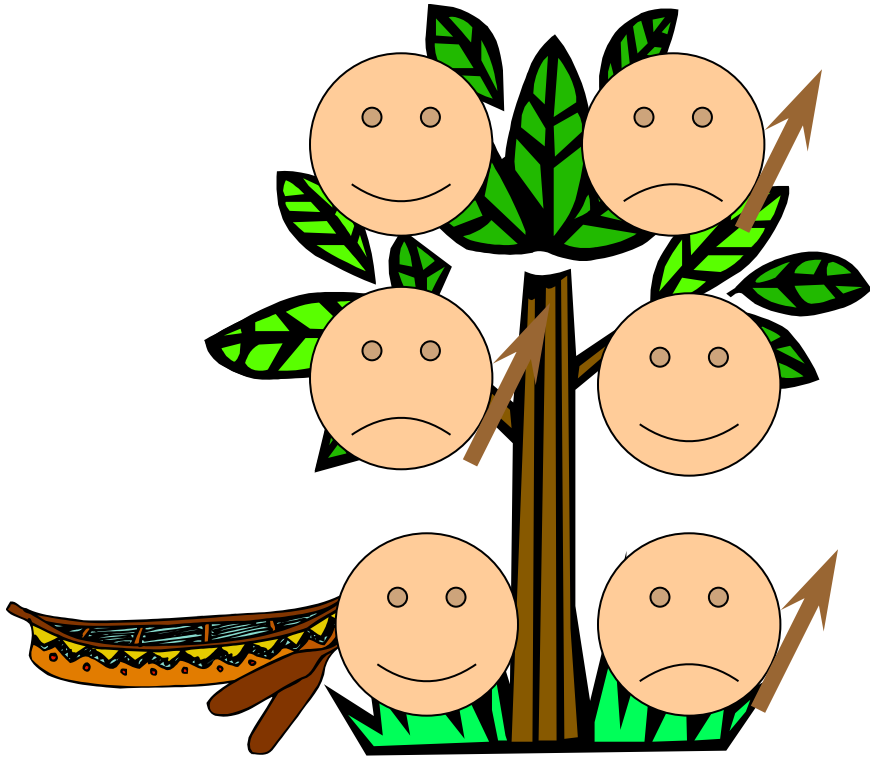


Credit: Khola Naseem

Missionaries and Cannibals

(0,0,0) : Goal State





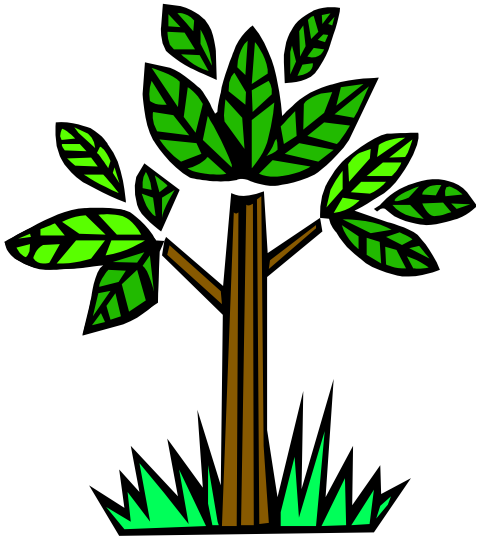
Credit: Khola Naseem

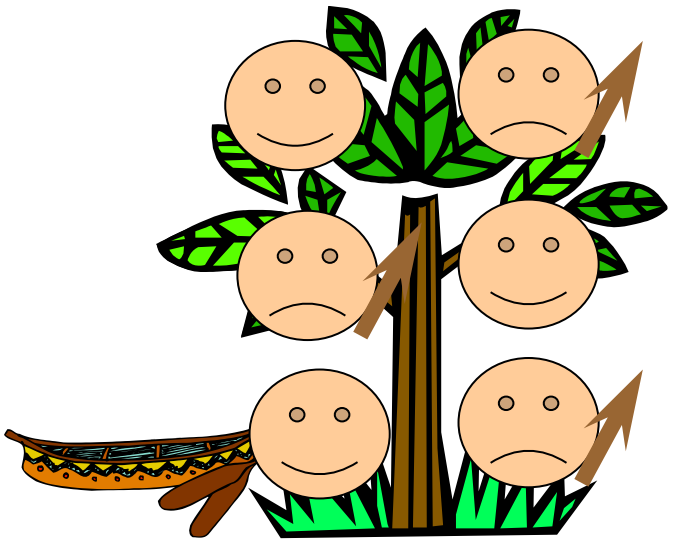
Missionaries and Cannibals

Solution = the sequence of actions within the path :

[(3,3,1)→ (2,2,0)→(3,2,1) →(3,0,0) →(3,1,1) →(1,1,0) →(2,2,1) →(0,2,0) →(0,3,1) →(0,1,0) → (0,2,1) →(0,0,0)]

Cost = 11 crossings





Credit: Khola Naseem

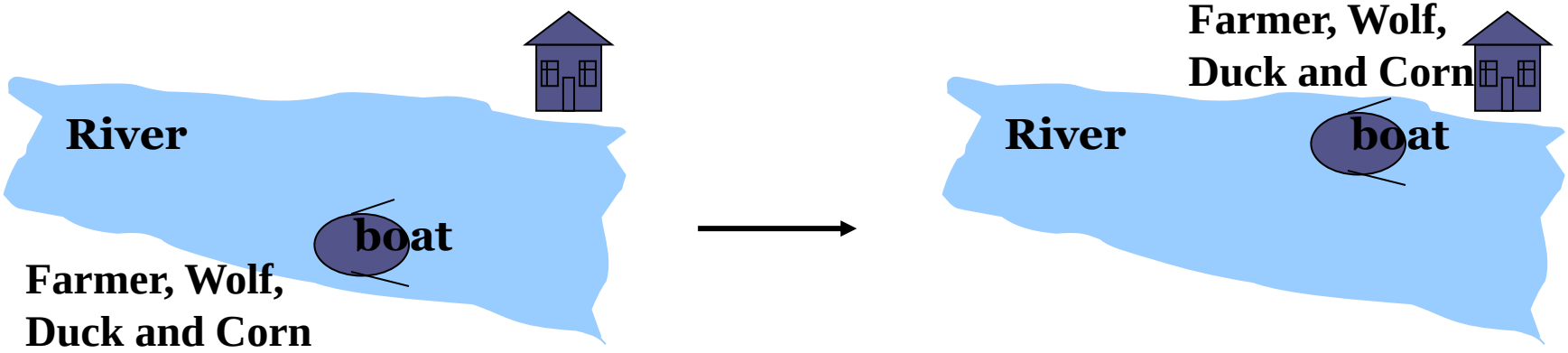
Missionaries and cannibals(Another possibility)

- States: five numbers (cl, ml, cr, mr, b) representing
  - # of missionaries and cannibals at left
  - the # of missionaries and cannibals at right
  - boats on the left bank of the river.
- Initial state: (3, 3, 0, 0, LEFT)
- **Operators:** in a given direction, take
  - one missionary,
  - one cannibal,
  - two missionaries,
  - two cannibals,
  - one missionary and one cannibal across the river
- Goal Test: reached state (0, 0, 3, 3, RIGHT)?
- Path Cost: Number of crossings.

Credit: Khola Naseem

The River Problem

- A farmer wishes to carry a wolf, a duck and corn across a river, from the south to the north shore. The farmer has a small rowing boat. The boat can only carry at most the farmer and one other item.
- If left unattended the wolf will eat the duck and the duck will eat the corn.



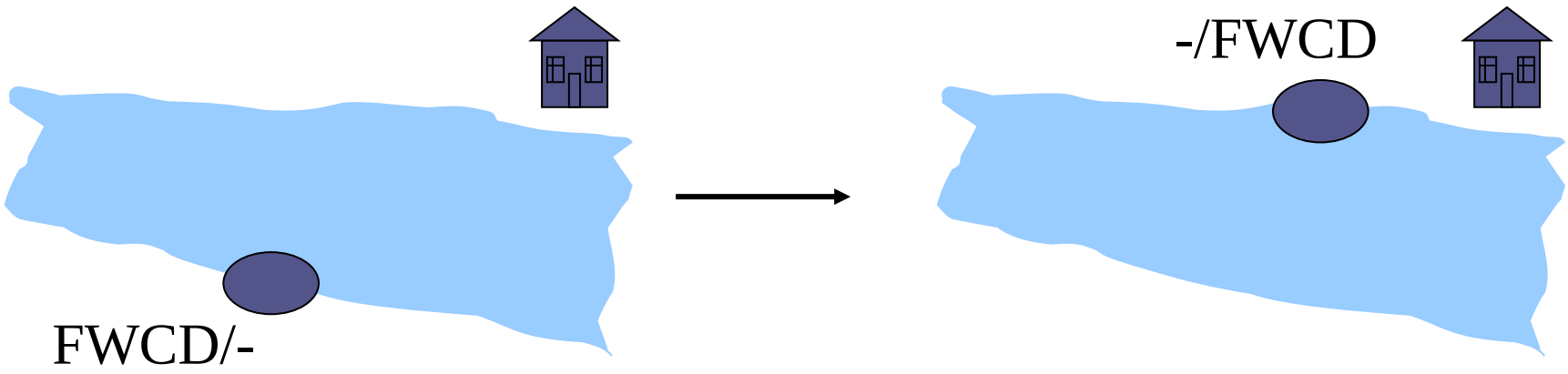
How can the farmer safely transport the wolf, the duck and the corn to the opposite shore?

Credit: Khola Naseem

The River Problem

- The River Problem:

F=Farmer W=Wolf D=Duck C=Corn /=River



How can the farmer safely transport the wolf, the duck and the corn to the opposite shore?

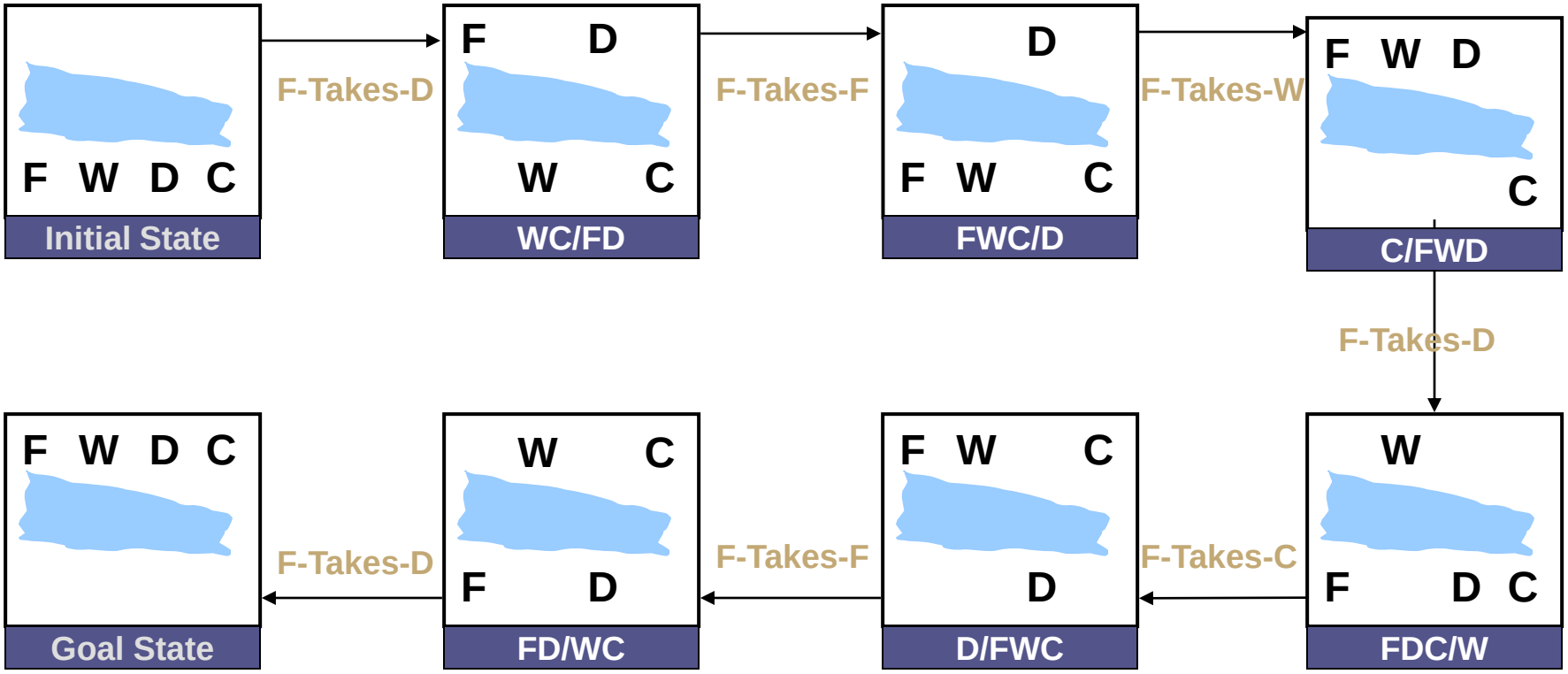
The River Problem

Problem formulation:

- State representation: location of farmer and items in both sides of river
  - [items in South shore / items in North shore] : (FWDC/-, FD/WC, C/FWD …)
- Initial State: farmer, wolf, duck and corn in the south shore
  - FWDC/-
- Goal State: farmer, duck and corn in the north shore
  - -/FWDC
- Operators: the farmer takes in the boat at most one item from one side to the other side
  (F-Takes-W, F-Takes-D, F-Takes-C, F-Takes-Self [himself only])
- Path cost: the number of crossings

The River Problem

- Problem solution: (path Cost = 7)
- While there are other possibilities here is one 7 step solution to the river problem



Credit: Khola Naseem

The Real world Problem(Chapter 3 page 74)

- The traveling salesperson problem (TSP) is a touring problem in which each city must be visited exactly once
- A VLSI layout problem requires positioning millions of components and connections on a chip
- Robot navigation
- Automatic assembly sequencing of complex objects by a robot
- Another important assembly problem is protein design, in which the goal is to find a sequence of amino acids that will fold into a three-dimensional protein with the right properties to cure some disease.

Credit: Khola Naseem