Design and Analysis of Algorithms

Project Report

Saadat Hussain (01-134211-078)
Shanze Malik (01-134211-085)

BSCS 5B
Bahria University Islamabad

Contents

1 Problem Statement

The Toads and Frogs puzzle, introduced in Levintin's book "Algorithmic Puzzles," is a classic game. It involves arranging tiles in a linear manner, where some are occupied by toads and others by frogs. The objective is to exchange the positions of the toads and frogs by adhering to a set of rules. The tiles are placed in a line, with toads and frogs alternating, and an empty tile separating them. A valid move involves swapping either a toad or a frog with the empty tile. The puzzle is considered solved when all the toads have successfully swapped places with the frogs, resulting in a new configuration where the frogs are on the left side and the toads are on the right side.

2 Objective

The initial setup of the puzzle consists of an array of 2n+1 (we have limited it to 9) elements, where n represents the number of toads or frogs on each side. The toads are represented by the number 1, and the frogs are represented by the number 2. The objective is to transform the initial configuration so that all the frogs are on the left side and all the toads are on the right side. The puzzle allows you to make moves by selecting a position on the board and moving the corresponding piece according to specific rules. A toad can jump over a frog and land on an empty space, while a frog can jump over a toad and land on an empty space. The goal is to find a sequence of moves that achieves the desired configuration with all the frogs on the left and all the toads on the right. The objective of the Toads and Frogs puzzle is to exercise problem-solving skills and logical thinking by finding a solution to the puzzle, considering the constraints and rules of movement.

3 Algorithm Design Technique

This algorithm uses **Brute Force** technique to compute the solution of the puzzle.

4 Algorithm Design

```
ALGORITHM ToadsAndFrogs //Brute Force Algorithm
Input: An array of 2n+1 elements with all the Toads(n) on the
left side and Frogs(n) on the right side
Output: An array of 2n+1 elements with all the Frogs(n) on the
left side and Toads(n) on the right side

PuzzleSizeLeft <- n
PuzzleSizeRight <- n
PuzzleSize <- 2n+1
CALL NewGame(PuzzleSizeLeft, PuzzleSizeRight)
WHILE NOT IsWin()
    INPUT index
    CALL TryMove(index)

PRINT ("You have solved the puzzle!")

//FUNCTIONS
FUNCTION NewGame(PuzzleSizeLeft, PuzzleSizeRight)
    Puzzle[PuzzleSizeLeft + PuzzleSizeRight + 1]
    // Initializing
    FOR i <- 0 TO PuzzleSizeLeft - 1 DO
        Puzzle[i] <- 1 //Toad
    Puzzle[PuzzleSizeLeft] <- 0
    FOR i <- PuzzleSizeLeft TO PuzzleSize DO
        Puzzle[i] <- 2 //Frog

FUNCTION TryMove(position)
    IF Puzzle[position] != 0 THEN
        IF Puzzle[position] = 1 THEN
            IF position + 1 < PuzzleSize AND Puzzle[position + 1]
            = 0 THEN
                Puzzle[position + 1] <- 1
                Puzzle[position] <- 0
            ELSE IF position + 2 < PuzzleSize AND
            Puzzle[position + 2] = 0 THEN
                Puzzle[position + 2] <- 1
                Puzzle[position] <- 0
        ELSE IF Puzzle[position] = 2 THEN
            IF position - 1 >= 0 AND Puzzle[position - 1] = 0
            THEN
                Puzzle[position - 1] <- 2
                Puzzle[position] <- 0
            ELSE IF position - 2 >= 0 AND Puzzle[position - 2]=0 THEN
                Puzzle[position - 2] <- 2
                Puzzle[position] <- 0

FUNCTION IsWin()
    win <- true
    FOR i <- 0 TO PuzzleSizeLeft - 1 DO
        IF Puzzle[i] != 2 THEN
```

```
        win <- false
IF Puzzle[PuzzleSizeLeft] != 0 THEN
    win <- false
FOR i <- PuzzleSizeLeft + 1 TO PuzzleSize DO
    IF Puzzle[i] != 1 THEN
        win <- false
RETURN win
```

5 Efficiency Analysis

$3 + \sum_{i=0}^{n} 1 + 1 + \sum_{n+1}^{2n+1} 1$
3 + (n-1 - 0 + 1) + 1 + (2n+1 - (n+1) +1)
3 + n + 1 + n + 2
2n + 6
Time Complexity = O(n)

6 Code

- **HTML**

```
<!DOCTYPE html>
<html>
<head>
  <title>Toads and Frogs</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div id="ToadsAndFrogs" class="container">
    <div id="About">
      <h1 id="Title"> Toads and Frogs Puzzle </h1>
      <p id="Instructions">
        <b>
          <div>
            Instructions:
      </p>
    </div>
    The goal of this Puzzle is for the Brown (Toads) and
    Green (Frogs) to change sides. Brown can only move
    right, and
    Green can only move left. <p>
      There are two legal moves:
      </b>
    <p> 1. A frog/toad can jump onto an adjacent free slot.
    </p>
    <p> 2. A frog/toad can jump over another frog/toad
    onto a free slot. </p>
    </p>
  </div>
  <canvas id="Canvas" onclick="canvasClick"></canvas>
  <form id="Settings">
    <label for="PuzzleSymmetry">Symmetric</label>
    <input id="PuzzleSymmetry" type="checkbox" checked
    onclick="PuzzleSymmetric()" />
    <label for="PuzzleSizeLeft" id="PuzzleSizeLeftLabel">
    Frogs on left</label>
    <input id="PuzzleSizeLeft" max="4" min="1" value="3"
    type="range" width="100" oninput="PuzzleSizeChanged()"
      onchange="PuzzleSizeChanged()" />
    <label for="PuzzleSizeRight" id="PuzzleSizeRightLabel">
    Frogs on right</label>
    <input id="PuzzleSizeRight" max="4" min="1" value="3"
    type="range" width="100" oninput="PuzzleSizeChanged()"
      onchange="PuzzleSizeChanged()" />
    <input id="PuzzleRestart" type="button" value="Restart"
    onclick="NewGame()" />
  </form>
  \h5 id="Author">by Saadat & Shanze</h5>
  </div>
```

```
<script src="main.js"></script>
</body>
</html}
```

- CSS

```
.container {
   width: 90%;
   margin: auto;
 }

h1
{
  text-align: center;
  color: brown;
}

 input[type='range'] {
   width: 90px;
 }

 label {
   display: block;
   font-family: system-ui, -apple-system,
   BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen,
   Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue',
   sans-serif;
   font-size: 11px;
   margin: 3px;
 }
 #About {
   font-family: system-ui, -apple-system,
   BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen,
   Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue',
   sans-serif;
 }
 #PuzzleRestart {
   display: Block;
   background: #ff0000;
   padding: 5px;
   margin: 3px;
 }

 #Author {
   font-family: system-ui, -apple-system,
   BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen,
   Ubuntu, Cantarell, 'Open Sans', 'Helvetica Neue',
   sans-serif, sans-serif;
   font-size: 16px;
   text-align: center;
 }
```

- **JavaScript**

```
// Global variables
var PuzzleSizeLeft = 3;
var PuzzleSizeRight = 3;
var Puzzle = [];
var MoveCount = 0;

//Driver
var c = document.getElementById("Canvas");
c.addEventListener("click", canvasClick);
document.getElementById("PuzzleSizeLeft").value = PuzzleSizeLeft;
document.getElementById("PuzzleSizeRight").value = PuzzleSizeRight;
requestAnimationFrame(Animation);
//PuzzleSymmetric();
NewGame();
function Animation()
{
    DrawCanvas();
    requestAnimationFrame(Animation);
}

function NewGame()
{
    MoveCount = 0;
    Puzzle = new Array(PuzzleSizeLeft + PuzzleSizeRight + 1);

    for (var i = 0; i < PuzzleSizeLeft; i++)    // Left
    {
        Puzzle[i] = 1;
    }
    Puzzle[PuzzleSizeLeft] = 0;                 // Middle
    for (var i = 0; i < PuzzleSizeRight; i++)    // Right
    {
        Puzzle[PuzzleSizeLeft + PuzzleSizeRight - i] = 2;
    }

    DrawCanvas();
}

function PuzzleSymmetric() {
    if (document.getElementById("PuzzleSymmetry").checked === true){
        document.getElementById("PuzzleSizeLeftLabel").innerHTML =
        "Frogs on either side";
        document.getElementById("PuzzleSizeRightLabel")
        .style.opacity = 0.4;
        document.getElementById("PuzzleSizeRight")
        .style.opacity = 0.4;
        document.getElementById("PuzzleSizeRight")
        .disabled = true;
    }
    document.getElementById("PuzzleSizeRight").value =
    document.getElementById("PuzzleSizeLeft").value;
```

```
    NewGame();
}

function TryMove(frog) {
    if (Puzzle[frog] != 0) // there is a frog there
    {
        if (Puzzle[frog] == 1) // moving right
        {
            if (frog + 1 < Puzzle.length && Puzzle[frog + 1] == 0){
                Puzzle[frog + 1] = 1;
                Puzzle[frog] = 0;
                MoveCount++;
            }
            else if (frog + 2 < Puzzle.length && Puzzle[frog + 2]
            == 0) {
                Puzzle[frog + 2] = 1;
                Puzzle[frog] = 0;
                MoveCount++;
            }
        }
        else if (Puzzle[frog] == 2) // moving left
        {
            if (frog - 1 >= 0 && Puzzle[frog - 1] == 0) {
                Puzzle[frog - 1] = 2;
                Puzzle[frog] = 0;
                MoveCount++;
            }
            else if (frog - 2 >= 0 && Puzzle[frog - 2] == 0) {
                Puzzle[frog - 2] = 2;
                Puzzle[frog] = 0;
                MoveCount++;
            }
        }
    }
}

function IsWin() {
    var won = true;     // assume victory
    for (var i = 0; i < PuzzleSizeRight; i++)    // Left
    {
        if (Puzzle[i] != 2)
            won = false;
    }
    if (Puzzle[PuzzleSizeRight] != 0)
        won = false;                          // Middle
    for (var i = 0; i < PuzzleSizeLeft; i++)    // Right
    {
        if (Puzzle[PuzzleSizeLeft + PuzzleSizeRight - i] != 1)
            won = false;
    }

    if (won) {
```

```
        alert("Congratulations! You solved the Puzzle in " +
        MoveCount + " moves.");
        NewGame();
    }
}

function canvasClick(event) {
    var rect = c.getBoundingClientRect();
    var clickX = event.pageX - rect.left
    var stone = Math.floor(clickX / (c.width / (PuzzleSizeLeft +
    PuzzleSizeRight + 1)));
    TryMove(stone);
    DrawCanvas();
    IsWin();
}

function PuzzleSizeChanged() {
    if (document.getElementById("PuzzleSymmetry").checked === true)
        document.getElementById("PuzzleSizeRight").value =
        document.getElementById("PuzzleSizeLeft").value;

    PuzzleSizeLeft = parseInt(document.getElementById("PuzzleSizeLeft").value);
    PuzzleSizeRight = parseInt(document.getElementById("PuzzleSizeRight").value);
    NewGame();
}
function DrawCanvas() {
    var appContainer = document.getElementById("ToadsAndFrogs");
    var appWidth = appContainer.offsetWidth;
    var stoneWidth = ((appWidth - 10) / ((PuzzleSizeLeft + PuzzleSizeRight) + 1));
    c.width = appWidth - 10;
    c.height = stoneWidth;
    var ctx = c.getContext("2d");
    ctx.clearRect(0, 0, c.width, c.height);
    for (var i = 0; i<(PuzzleSizeLeft+PuzzleSizeRight)+1;i++){
        // Draw boxes
        ctx.fillStyle = "#ffffff";
        ctx.fillRect(i * stoneWidth, 0, stoneWidth, stoneWidth);

        // Draw "frogs"
        if (Puzzle[i] != 0) {
            var frogImg = new Image();
            if (Puzzle[i] === 1)
                frogImg.src = "frog2.png";
            else
                frogImg.src = "frog1.png";
            ctx.drawImage(frogImg, i * stoneWidth, 0, stoneWidth, stoneWidth);
        }
    }
}
```

7 Output

Toads and Frogs Puzzle

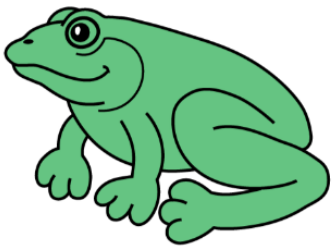
Instructions:

The goal of this puzzle is for the Brown (Toads) and Green (Frogs) to change sides. Brown can only move right, and Green can only move left.

There are two legal moves:

1. A frog/toad can slide into an ajacent free slot.

2. A frog/toad can jump over another frog/toad into a free slot.



Toads and Frogs Puzzle

Instructions:

The goal of this puzzle is for the Brown (Toads) and Green (Frogs) to change sides. Brown can only move right, and Green can only move left.

There are two legal moves:

1. A frog/toad can slide into an ajacent free slot.

2. A frog/toad can jump over another frog/toad into a free slot.

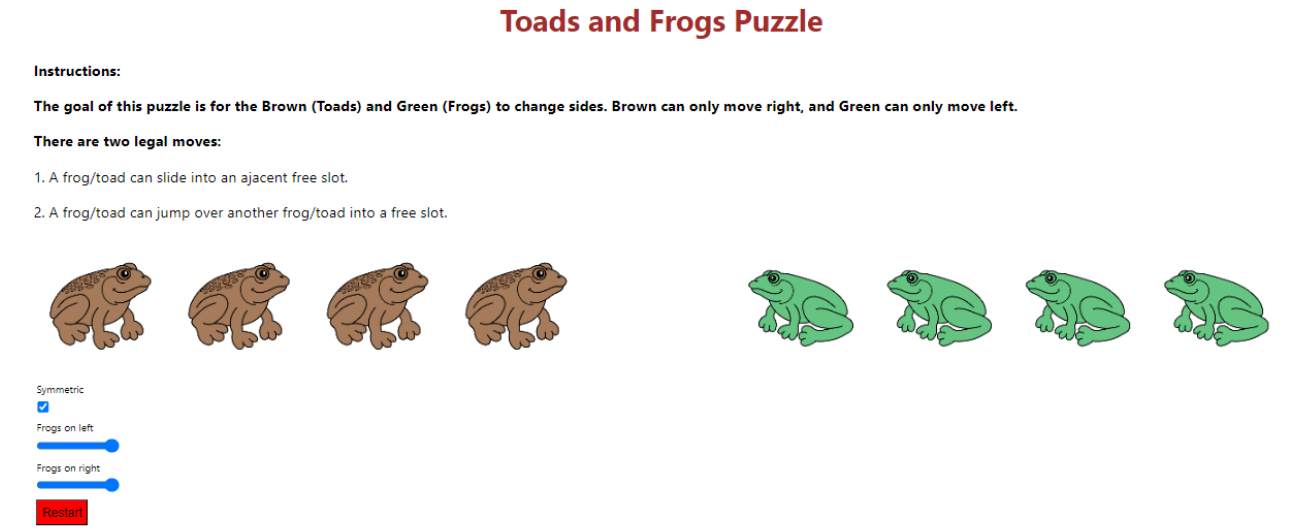
Symmetric

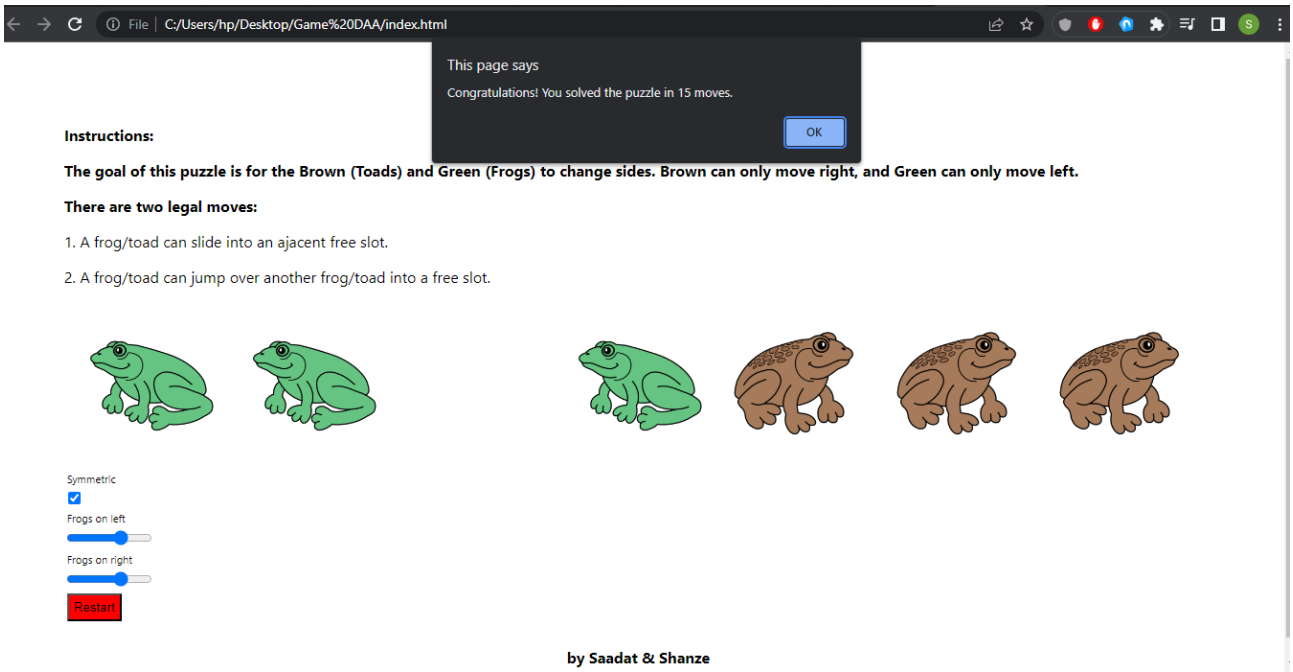
Frogs on left

Frogs on right

Restart

by Saadat & Shanze





8 Conclusion

In conclusion, the Toads and Frogs game is a challenging puzzle that requires strategic thinking and problem-solving skills.The game presents an opportunity to exercise logical reasoning and planning. Players must carefully analyze the current board state, consider the available moves, and anticipate the consequences of their actions. Finding an optimal solution requires thoughtful decision-making and considering multiple possibilities.Toads and Frogs game is an excellent way to challenge oneself and improve logical reasoning skills while enjoying the process of solving a captivating puzzle.