1)

1. A linear search must iterate one-by-one through the list, which means the search might potentially have to check every single index before it finds the result. Conversely, a binary search usually eliminates multiple possibilities with each iteration.
2. If the list is not already sorted by value, or if you don’t know whether the list is sorted or not.

2) *You do not have to use all the rows… stop once you have found the number you are looking for*

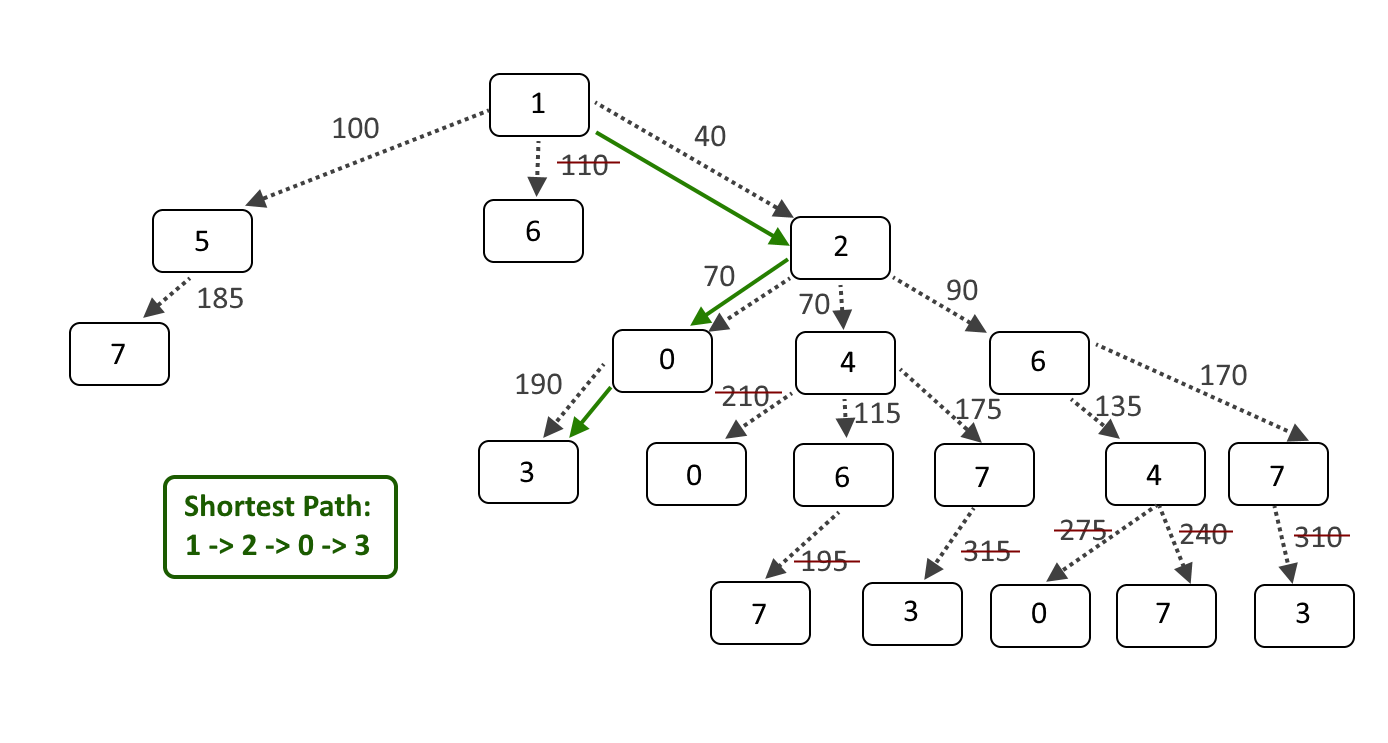
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Step*** | ***minLocation*** | ***maxLocation*** | ***middleLocation*** | ***middleValue*** |
| *1* | *1* | *10* | *5* | *34* |
| *2* | *1* | *4* | *2* | *8* |
| *3* | *3* | *4* | *3* | *12* |
| *4* | *4* | *4* | *4* | *19* |
| *5* |  |  |  |  |

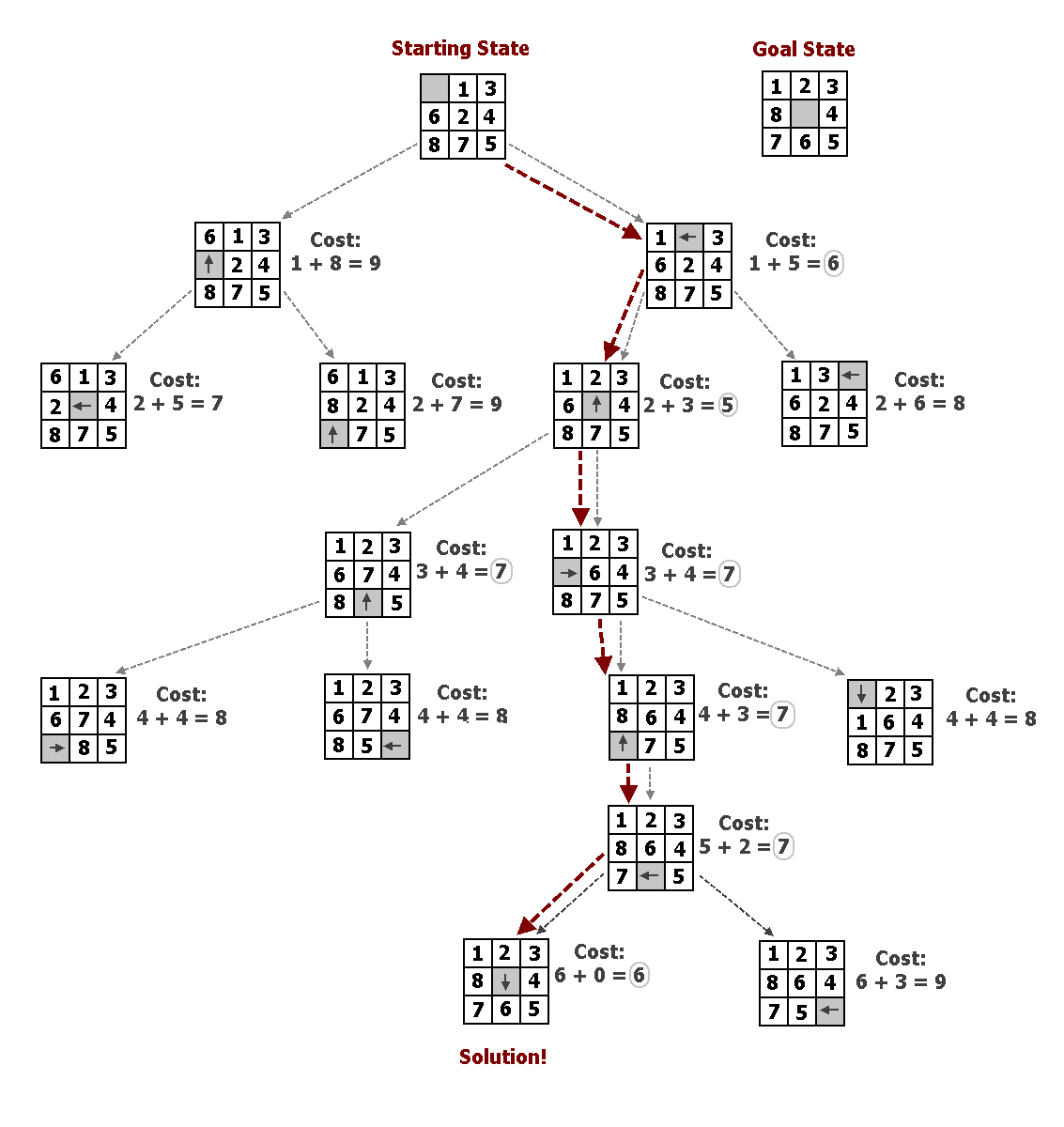
3A) *You SHOULD be using each row in this table. Make sure to record steps even where nothing visibly changes.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STEP** | 5 | 3 | 9 | 8 | 4 | 2 | 6 | 7 |
| **1** | 3 | 5 | 9 | 8 | 4 | 2 | 6 | 7 |
| **2** | 3 | 5 | 9 | 8 | 4 | 2 | 6 | 7 |
| **3** | 3 | 5 | 8 | 9 | 4 | 2 | 6 | 7 |
| **4** | 3 | 4 | 5 | 8 | 9 | 2 | 6 | 7 |
| **5** | 2 | 3 | 4 | 5 | 8 | 9 | 6 | 7 |
| **6** | 2 | 3 | 4 | 5 | 6 | 8 | 9 | 7 |
| **7** | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

3B) *You SHOULD be using each row in this table. Make sure to record steps even where nothing visibly changes.*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **STEP** | 5 | 3 | 9 | 8 | 4 | 2 | 6 | 7 |
| **1** | 2 | 3 | 9 | 8 | 4 | 5 | 6 | 7 |
| **2** | 2 | 3 | 9 | 8 | 4 | 5 | 6 | 7 |
| **3** | 2 | 3 | 4 | 8 | 9 | 5 | 6 | 7 |
| **4** | 2 | 3 | 4 | 5 | 9 | 8 | 6 | 7 |
| **5** | 2 | 3 | 4 | 5 | 6 | 8 | 9 | 7 |
| **6** | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 8 |
| **7** | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

4) **

5) **

6)

|  |
| --- |
| *onEvent("button\_roll", "click", function( ) {*  *setText("label\_die1", randomNumber(1, 6));*  *setText("label\_die2", randomNumber(1, 6));*  *setText("label\_die3", randomNumber(1, 6));*    *// Compute new score & update label text*  *updateScore();*  *});*  *function updateScore()*  *{*  *// Points earned for this roll*  *var scoreThisRoll = 0;*    *// Check for combinations*  *if ((getNumber("label\_die1") == getNumber("label\_die2")) && (getNumber("label\_die1") == getNumber("label\_die3"))) {*  *scoreThisRoll = 50; // If all dice are equal, that's worth 50 points*  *setText("text\_area\_eventLog","ROLLED TRIPLES!!! Won "+scoreThisRoll+" points.\n"+getText("text\_area\_eventLog"));*  *} else if ((getNumber("label\_die1") == getNumber("label\_die2")) || (getNumber("label\_die1") == getNumber("label\_die3")) || (getNumber("label\_die2") == getNumber("label\_die3"))){*  *scoreThisRoll = 25; // If they're not all equal, but two dice match, that's worth 25 points*  *setText("text\_area\_eventLog","Rolled doubles!! Won "+scoreThisRoll+" points.\n"+getText("text\_area\_eventLog"));*  *} else {*  *// If no bonus, the score is the sum of all dice*  *scoreThisRoll = getNumber("label\_die1") + getNumber("label\_die2") + getNumber("label\_die3");*  *setText("text\_area\_eventLog","Rolled dice. Won "+scoreThisRoll+" points.\n"+getText("text\_area\_eventLog"));*  *}*    *// Update score display*  *setText("label\_ScoreValue",getNumber("label\_ScoreValue")+scoreThisRoll);*  *}* |

7)

|  |
| --- |
| *var randButtonID;*  *var currentPlayer = 1;*  *var playerScore=[0,0];*  *var scoreWinThreshold = 5;*  *var minColorVariance = 20;*  *var maxColorVariance = 30;*  *var startTime;*  *setGame();*  *// Start a brand new game*  *function setGame()*  *{*  *// Reset values*  *currentPlayer = 1;*  *playerScore = [0,0];*  *startTime = getTime();*  *minColorVariance = maxColorVariance-10;*    *// Reset visual elements*  *refreshScoreTextLabels();*  *hideElement("player1Win\_label");*  *hideElement("player2Win\_label");*    *// Reset game board*  *setBoard();*  *}*  *// Re-Randomize the buttons*  *function setBoard()*  *{*  *var randColorR = randomNumber(maxColorVariance,255-maxColorVariance);*  *var randColorG = randomNumber(maxColorVariance,255-maxColorVariance);*  *var randColorB = randomNumber(maxColorVariance,255-maxColorVariance);*  *var colorVariance = ((randomNumber(0,1) == 1) ? randomNumber(minColorVariance,maxColorVariance) : randomNumber(-minColorVariance,-maxColorVariance));*  *var color = rgb(randColorR, randColorG, randColorB);*  *var diffColor = rgb(randColorR+colorVariance,randColorG+colorVariance,randColorB+colorVariance);*  *randButtonID = "button"+randomNumber(1,4);*  *setProperty("button1", "background-color", color);*  *setProperty("button2", "background-color", color);*  *setProperty("button3", "background-color", color);*  *setProperty("button4", "background-color", color);*  *setProperty(randButtonID, "background-color", diffColor);*  *console.log("The correct button is "+randButtonID);*  *}*  *// Compute new score and refresh the score display text*  *function updateScoreBy(amt)*  *{*  *// Update current player's score with the given value*  *playerScore[currentPlayer-1]+=amt;*    *// Update text*  *refreshScoreTextLabels();*  *console.log("Player "+currentPlayer+" Score: "+playerScore[currentPlayer-1]);*  *}*  *function refreshScoreTextLabels()*  *{*  *setText("score1\_label",playerScore[0]);*  *setText("score2\_label",playerScore[1]);*  *}*  *function switchPlayer(){*  *// Hide current player's highlight*  *var currentPlayerHighlightID = "player"+currentPlayer+"\_highlight";*  *hideElement(currentPlayerHighlightID);*  *// Swap players*  *currentPlayer = ((currentPlayer == 1) ? 2 : 1);*  *// Show the new player's highlight*  *currentPlayerHighlightID = "player"+currentPlayer+"\_highlight";*  *showElement(currentPlayerHighlightID);*  *console.log("Current player: "+currentPlayer);*  *}*  *function checkCorrect(buttonID)*  *{*  *console.log("Checking "+buttonID+"...");*    *if (buttonID == randButtonID) {*  *updateScoreBy(1); // Correct*  *} else {*  *updateScoreBy(-3); // Incorrect*  *}*    *// Check if that player won*  *if (playerScore[currentPlayer-1] >= scoreWinThreshold)*  *{*  *// Display how long the game lasted (while avoiding division by zero, just in case)*  *var totalGameTime = (getTime()-startTime);*  *setText("label\_gameDurationLabel","That game lasted for "+(totalGameTime != 0 ? totalGameTime/1000 : 0)+" seconds!");*    *// Declare the winner*  *console.log("Player "+currentPlayer+" wins!!!");*  *showElement("player"+currentPlayer+"Win\_label");*  *setScreen("gameOver\_screen");*  *}*  *else*  *{*  *// If the player didn't win, do another round*  *switchPlayer();*  *setBoard();*  *}*  *}*  *onEvent("button1", "click", function( ) {*  *console.log("button1 clicked!");*  *checkCorrect("button1");*  *});*  *onEvent("button2", "click", function( ) {*  *console.log("button2 clicked!");*  *checkCorrect("button2");*  *});*  *onEvent("button3", "click", function( ) {*  *console.log("button3 clicked!");*  *checkCorrect("button3");*  *});*  *onEvent("button4", "click", function( ) {*  *console.log("button4 clicked!");*  *checkCorrect("button4");*  *});*  *onEvent("button\_playAgain", "click", function( ) {*  *setScreen("gameWelcome\_screen");*  *});*  *onEvent("button\_newGameEasy", "click", function( ) {*  *maxColorVariance = 40;*  *setGame();*  *setScreen("gamePlay\_screen");*  *});*  *onEvent("button\_newGameMedium", "click", function( ) {*  *maxColorVariance = 30;*  *setGame();*  *setScreen("gamePlay\_screen");*  *});*  *onEvent("button\_newGameHard", "click", function( ) {*  *maxColorVariance = 20;*  *setGame();*  *setScreen("gamePlay\_screen");*  *});* |