# Unit-I

Introduction – Growth Rates of Various Functions and History of AI
Problem Solving – State Space Search: Exhaustive and Heuristic Search
Puzzles and Games Playing

Growth Rates of Various Functions

| S. No | Class | Terminology | Example | Tractable P Problems/ Intractable NP Complete Problems |
|---|---|---|---|---|
| 1 | 1 | Constant growth | Finding midpoint of an array | Tractable, P Problems |
| 2 | log(n) | Logarithmic growth | Binary Search | |
| 3 | n | Linear growth | Linear Search | |
| 4 | n.log(n) | | Merge Sort | |
| 5 | $n^2$ | Quadratic growth | Insertion Sort | |
| 6 | $n^3$ | Cubic growth | Seeing if an element appears 3 times in a list | |
| 7 | $n^c$ | Polynomial growth | The two above | |
| 8 | $2^n$ | Exponential growth | Towers of Hanoi | Intractable, NP Complete Problems |
| 9 | $n^{n-2}$ | | No. of spanning trees generated from a graph G of n vertices | |
| 10 | n! | Factorial growth | Traveling Salesman Problem | |
| 11 | $2^m$<br>$m = 2^n$ | | Boolean Function of degree n | NP-Hard |

An algorithm with growth rate equal to or larger than exponential growth is called ‘Intractable’, because, for even moderate input size, its run time is impractically long.

Problems from Graph Theory

| S. No | Polynomial-Bounded Algorithms | Run-Time Bounds<br>n: no. of vertices<br>e: no. of edges | | Non-Polynomial Bounded Algorithms |
|---|---|---|---|---|
| 1 | Connectedness and components | $n^2$ or e | | Chromatic Number |
| 2 | Spanning Tree | e | | Smallest Dominant Set |
| 3 | Minimal Spanning Tree | $n^2$ | | Maximal Clique |
| 4 | Fundamental Circuit-Set | $n^v$ $2 \leq v \leq 3$ | | Hamiltonian Circuit |
| 5 | Cut-Vertices and Blocks | $n^2$ or e | | Directed Hamiltonian Circuit |
| 6 | Bridges | $n^2$ or e | | Traveling Salesman Problem |
| 7 | Shortest Path between two vertices | $n^2$ | | Minimal Feedback Edge-Set |
| 8 | Transitive Closure | $n^\alpha$ $2 \leq \alpha \leq 3$ | | Minimal Feedback Vertex-Set |
| 9 | Strong Connectedness and Fragments | $n^2$ or e | | Steiner Tree |
| 10 | Planarity | e | | Isomorphism |
| 11 | Topological Sorting | e | | |
| 12 | Maximal matching in a bipartite graph | $n^{5/2}$ | | |
| 13 | Minimal Cut | $n^\beta$ $2 \leq \beta \leq 3$ | | |
| 14 | Minimal Edge Cover | $n^3$ | | |
| $t \leq \alpha n^k$ or $t \leq \beta e^q$ | | | | |

**Unit – I**

# Introduction

1. Some Definitions of Artificial Intelligence

   **<u>Artificial Intelligence</u>** is the study of how to make computers to do things which, at the moment, people do better (Rich and Knight).

   **<u>Agent</u>** is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. Performance measure gives the criteria that determine how successful an agent is.

   **<u>Ideal Rational Agent</u>**: For each possible percept sequence, an ideal rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence (everything that the agent has perceived so far) and whatever built-in knowledge the agent has.

   **<u>Cognitive Science</u>**: Interdisciplinary field which brings together computer models from AI and experimental techniques from Psychology that attempts to construct precise and testable theories of the workings of the human mind.

   **<u>Turing Test</u>**: (by Alan Turing, 1950) Ability to achieve human-level performance in all cognitive tasks, sufficient to fool an interrogator.

Systems that

| <u>Think like Humans</u> | <u>Think Rationally</u> |
|---|---|
| **Cognitive Science**<br>Hugeland, 1985: General Problem Solver.<br>Bellman, 1978: The automation of activities that we associate with human thinking, activities such as decision-making, problem solving, learning … | **Logic**<br>Charnaik and McDermott, 1985: The study of mental faculties through the use of computational models.<br>Winston, 1992: The study of the computations that make it possible to perceive, reason, and act. |
| <u>Act like Humans</u><br>**Turing Test**<br>Kurzweil, 1990: The art of creating machines that perform functions that require intelligence when performed by people.<br>Rich and Knight, 1991: The study of how to make computers do things which, at the moment, people do better. | <u>Act Rationally</u><br>**Agent**<br>Schalkoff, 1990: A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes.<br>Luger and Stubblefield, 1993: The branch of computer science that is concerned with automation of intelligent behavior. |

Ref: Rich & Knight.

1. A Historical Trace of Artificial Intelligence

1943-1956: The gestation of Artificial Intelligence

| | | |
|---|---|---|
| 1943 | Warren McCulloch and Walter Pitts | First AI Work |
| 1949 | Donald Hebb | Updating Rule |
| 1950 | Claude Shannon | Chess Programs |
| 1951 | Marvin Minsky and Dean Edmonds | First Neural Network Computer |
| 1953 | Alan Turing | Chess Programs |
| | McCarthy | Named the field as Artificial Intelligence (AI) |
| | Allen Newell and Herbert Simon | |

1952-1975: Early Enthusiasm and Great Expectations

| | | |
|---|---|---|
| 1952 | Arthur Samuel | Checkers |
| 1958 | MIT: Hohn McCarthy | LISP: Programs with Common Sense |
| 1958 | Marvin Minsky | Microworlds: Block World |
| 1959 | IBM: Nathaniel Rochester | |
| 1959 | Herbert Gelernter | Geometric Theorem Prover |
| 1960 | Widrow and Hoff | Enhanced Hebb's Learning Methods |
| 1962 | Widrow | Adalines (Networks) |
| 1962 | Frank Rosenblatt | Perceptrons |
| 1963 | James Slagle | SAINT Program: Closed Form Integration |
| 1963 | Winograd and Cowan | Lange number of elements represent and individual concept |
| 1967 | Daniel Bobrow | STUDENT Program: Algebra Story Problem |
| 1968 | Tom Evan | ANALOGY Program: Geometric Analogy Problems |
| 1970 | Patrick Winston | Learning Theory |
| 1971 | David Huffman | Vision Project |
| 1972 | Terry Winograd | Natural Language Understanding |
| 1974 | Scott Fahlman | Planner |
| 1975 | David Waltz | Constraint Propagation |

1966-1974: A Dose of Reality

| | | |
|---|---|---|
| <br>1958<br>1957<br><br>1973 | <br>Friedberg et. al<br>National Research Council<br>Lighthill | AI attempted Intractable Problems:<br>Machine Evolution (Genetic Algorithms)<br>English Translation of Russian Scientific Papers in the wake of the Sputnik Launch<br>Lighthill Report: Basis for decision to end British Government support to AI research as AI could not tackle 'Combinatorial Explosion'. |
| 1969 | Minsky and Papert | Book on 'Perceptrons': Proved that although Perceptrons could be shown to learn anything they were capable of representing, they could represent very little. |
| 1965 | Weizenbaum | Programs contained little or no knowledge of their subject matter.<br>ELIZA Program |

1969-1979: Knowledge-Based Systems

| | | |
|---|---|---|
| 1969 | Buchanan et al., | Weak Methods:<br>General Purpose Mechanisms trying to string together elementary reasoning steps to find complete solutions.<br>DENDRAL Program (First Knowledge Intensive System) |
| 1979<br>1977<br>1981<br>1983 | Duda et al.,<br>Schank and Abelson<br>Schank and Riesbeck<br>Dyer | Expert Systems:<br>Feigenbaum, Buchanan and Edward Shortlife: MYCIN Medical Diagnosis for blood infections using 450 rules.<br>PROSPECTOR: Exploratory Drilling at a geological site.<br>Understanding Natural Languages.<br>Understanding Natural Languages.<br>Understanding Natural Languages. |
| 1975 | Minsky | Frames:<br>Structured approach collecting together facts about particular object and event types, and arranging the types into a large taxonomic hierarchy analogous to a biological taxanomy |

1980-1988: AI becomes an Industry

| | | |
|---|---|---|
| 1982 | Mc Dermott | The first successful Commercial Expert System R1, began operation at Digital Equipment Corporation<br>A few million $ in 1980 to 2 billion $ by 1988 |

1986 – Present: The return of Neural Networks

| | | |
|---|---|---|
| 1969 | Bryson and Ho | Back-Propagation Learning |
| 1982 | Hopfield | Storage and Optimization properties of Networks |
| 1986 | Rumelhart and McClelland | Parallel and Distributed Processing of Back-Propagation |

1987- Present: Recent Events

| | | |
|---|---|---|
| | | Hidden Markov Models |
| 1977<br>1987 | Austin Tate<br>David Chapman | Simple Framework for Planning Programs |
| 1988 | Judea Pearl | Probabilistic Reasoning in Intelligent Systems: Acceptance of Probability and Decision Theory to AI. |
| 1982<br>1986 | Judea Pearl<br>Horvitz et al., | Belief Networks: Reasoning about the combination of uncertain evidence.<br>Normative Expert Systems: One that act rationally according to the laws of Decision Theory and do not try to imitate human experts. |

2. Problem Solving: Solution Trace in State-Space and in State-Transition Diagram

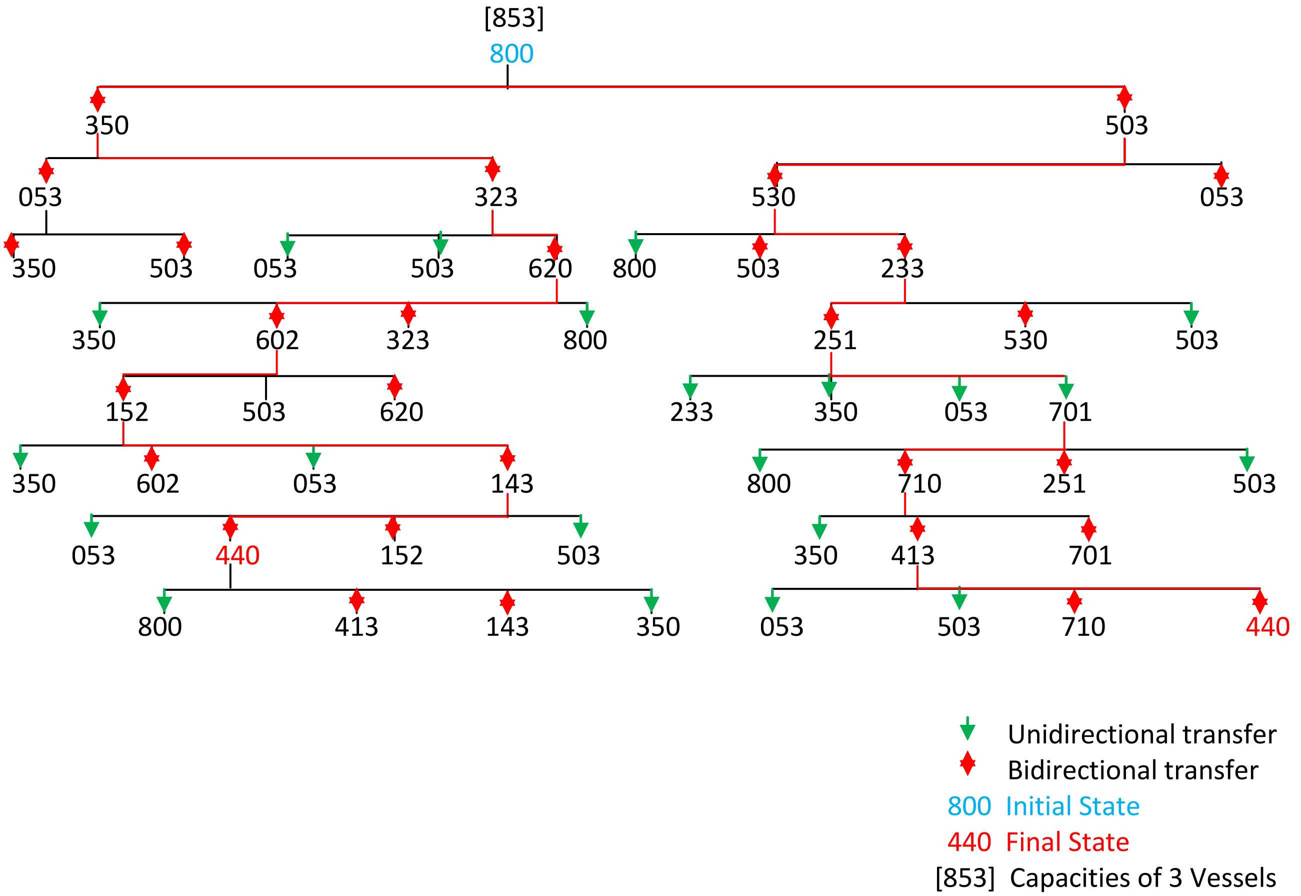



Figure 1. Problem Solving using State Space Approach: 853 Water Vessels Problem – Exhaustive Search & Heuristic Search©

V. Ashwini Kumar 18 March 2014

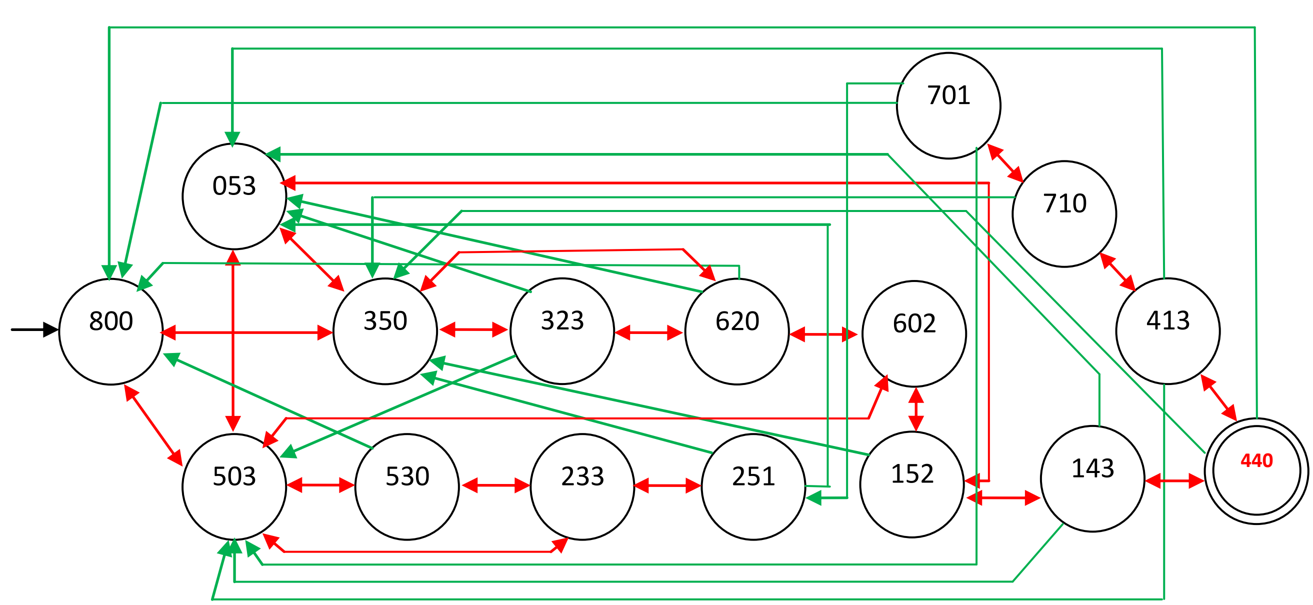



Figure 2. Problem Solving using State Transition Approach: 853 Water Vessels Problem – Exhaustive Search & Heuristic Search©

V. Ashwini Kumar 18 March 2014

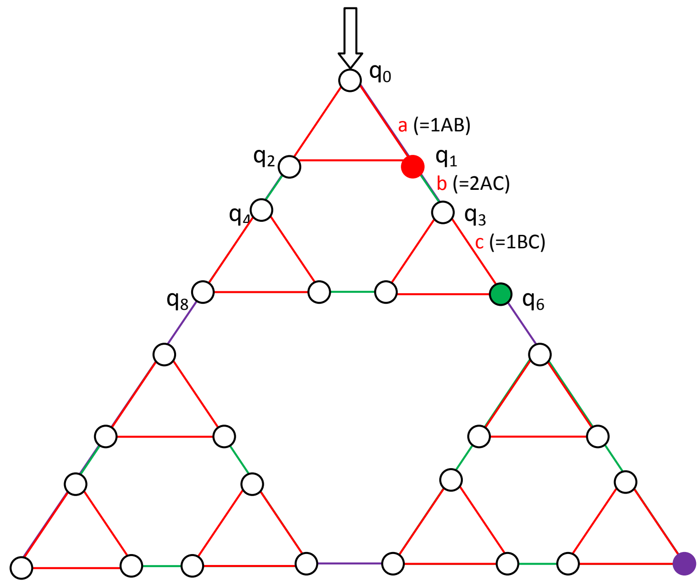



Figure 3. Illustration of Recursion represented through State Transition Diagram for Towers of Hanoi©.

Final States: 3 Pole – 3 Disc Case, 3 Pole – 2 Disc Case and 3 Pole – 1 Disc Case

Heuristic for Solution Path: Follow Recursion as shown above along the right most edge a-b-c of the Triangle.

V. Ashwini Kumar, Recursion Heuristic Search and Exhaustive Search in Towers of Hanoi

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | | | | 1 |
| 2 | - | - | | 2 | 1 | - | | - | 1 | 2 | | - | - | 2 |
| A | B | C | | A | B | C | | A | B | C | | A | B | C |
| $q_0$ | | | a | $q_1$ | | | b | $q_3$ | | | c | $q_6$ | | |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 1 |
| 2 | | | | 2 | | | | | | | | | 1 | | | | 1 | | | | | | | | | 2 | | | | 2 |
| 3 | - | - | | 3 | - | 1 | | 3 | 2 | 1 | | 3 | 2 | - | | - | 2 | 3 | | 1 | 2 | 3 | | 1 | - | 3 | | - | - | 3 |
| A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C |

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | 2 | | | | | | | | | | | | | | | | | | | | | | | | | 1 | | | | 1 | |
| 3 | | | | 3 | | | | 3 | | | | 3 | | 1 | | | | 1 | | 1 | | | | 1 | 2 | | | | 2 | | | | 2 | |
| 4 | - | - | | 4 | 1 | | | 4 | 1 | 2 | | 4 | - | 2 | | 4 | 3 | 2 | | 4 | 3 | 2 | | 4 | 3 | - | | 4 | 3 | - | | - | 3 | 4 |
| A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C | | A | B | C |

Figure 4. Heuristic Search: Recursive Sequence of Transitions for 2, 3 and 4 disc cases of Towers of Hanoi©.

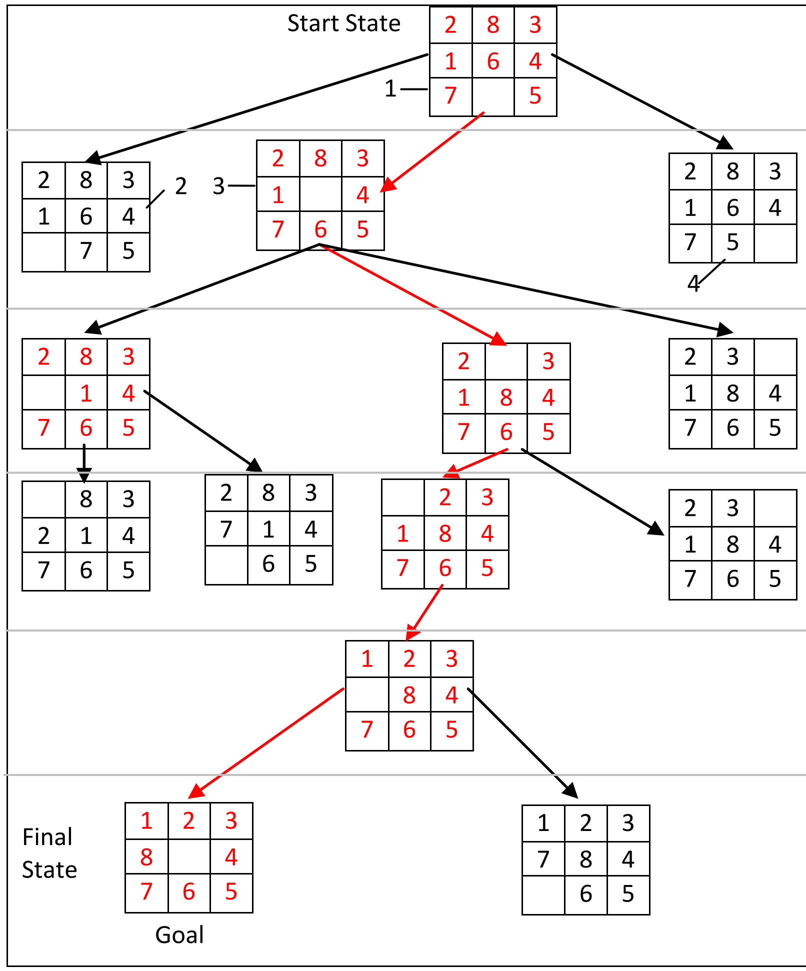
Number of Transitions required to reach final state is given by: $H_n = 2\ H_{n-1} + 1 = 2^n - 1$.

Where $n$ is the number of discs. If n = 64, $H_n$ = 18, 446, 744, 073, 709, 551, 615

## Heuristic Search — Eight Slide Puzzle Problem

**Heuristic**: A technique that improves the average-case performance on a problem-solving task, but does not necessarily improve the worst-case performance.

**Heuristic Function**: f(n): A function that help decide which node is the best one to expand next. This is a measure of the goodness of a state. When written as sum of g(n), the depth factor and h(n), the heuristic evaluation of a node help to explore promising paths to the goal, in this case, it is the number of tiles out of place (compared to goal state).




1. $f(n) = g(n) + h(n) = 0 + 5 = 5$
2. $f(n) = g(n) + h(n) = 1 + 6 = 7$
3. $f(n) = g(n) + h(n) = 1 + 3 = 4$
4. $f(n) = g(n) + h(n) = 1 + 6 = 7$
5. $f(n) = g(n) + h(n) = 2 + 4 = 6$
6. $f(n) = g(n) + h(n) = 2 + 4 = 6$
7. $f(n) = g(n) + h(n) = 2 + 5 = 7$
8. $f(n) = g(n) + h(n) = 3 + 4 = 7$
9. $f(n) = g(n) + h(n) = 3 + 5 = 8$
10. $f(n) = g(n) + h(n) = 3 + 3 = 6$
11. $f(n) = g(n) + h(n) = 3 + 5 = 8$
12. $f(n) = g(n) + h(n) = 4 + 2 = 6$
13. $f(n) = g(n) + h(n) = 5 + 0 = 5$
14. $f(n) = g(n) + h(n) = 5 + 3 = 8$

Figure 5. Heuristic Function evaluation to decide upon expansion of next nodes in State-Space of 8-Slide Puzzle Problem.

Algorithm: *Heuristic for 8-Slide Puzzle Problem*

1. Test h(n) = 0? If true go to 5, else go to 2.
2. Expand Nodes (States) in the next level of the Tree (breadth-wise).
3. Find f(n) for the above Nodes and compare to find the Node with minimum value.
4. Expand Node/s with minimum value.
   Repeat 1 to 4 until Final (goal) State with f(n) is Minimum and h(n) = 0.
5. End.

Heuristics in Puzzles and Games

9 Side-Flipping Tiles Problem – SFT9

Each tile the board is numbered from 1 to 9. Tiles can be flipped side-ways on to the next adjacent tile. Any one tile can be flipped at a time. Two or more tiles at a location is called compound tile. A compound tile cannot be flipped. Face value of a compound tile is the value of the tile at the top. Value of the board is the sum of the values of tiles seen on the board. In this case, it is 1 + 2 + . . . + 9 = 45.

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

For example, after the tiles with values 1, 3, 5, 7 and 9 are flipped side-ways, we get a state as shown in the figure. The value of the board is 1 + 3 + 5 + 9 = 18.

| | | |
|---|---|---|
| | 1 | |
| 5 | | 3 |
| | 9 | |

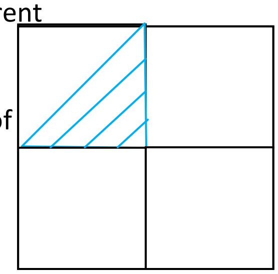
After the tiles 1, 2, 3, 8 and 3 are flipped side-ways, we get a state as shown in the figure. The value of the board is 1 + 2 + 3 + 7 = 13.

| | | |
|---|---|---|
| | | |
| 1 | 2 | |
| 7 | | 3 |

How do we get to a minimum value of the board with minimum number of flips?
Will this heuristic work for board with random sequence of numbers (1-9) as its initial state?

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

Triangular lamina in blue colour (hatched lines) has two sides with two different numbers. Say, 1 and 2 respectively. Similarly, eight triangles on the background square have different unique non-repeated numbers.
Triangular lamina in blue colour (hatched lines) can be made to flip on any of to occupy next adjacent triangular area in the square on each flip.
The value of the triangular area in the square gets updated when the lamina in blue colour leaves the space it occupied earlier.
How to get to a final state from a given state? State the update rule.
Is there a heuristic to improve search? State.



**Analyze Tic-Tac-Toe Game and Present the Strategies and the corresponding Heuristics to win the Game**

Types of Behavior (Strategy and Heuristic) of Players:

1. Play uniformly randomly (Unresponsive to opponent's actions) – Generally losing the game
2. Play to block in every move – Generally Drawing the game
3. Play to win by not blocking, but blocking only in a losing situation – Some-times winning the game, when the opponent's behavior is of type 1 or 3.

Assumption: the players do not change the strategies during the game.

Types of games:

| | | |
|---|---|---|
| 1-1 | 2-1 | 3-1 |
| 1-2 | 2-2 | 3-2 |
| 1-3 | 2-3 | 3-3 |

Notation: 11 means first player (1) playing the first (1) move. 24 means second player making the fourth move.

Type of Game: 3-3 (First move of the first player cannot be the cell 5, this belongs to Game Type 2)
The second player will win even if the first player marks in cell/position 3 or 9 instead of in cell/position 2.

| | | |
|---|---|---|
| 11<br>1 | <br>2 | 13<br>3 |
| 12<br>4 | 23<br>5 | <br>6 |
| 22<br>7 | 21<br>8 | 24<br>9 |

The second player will win even if the first player marks in cell/position 8 or 9 instead of in cell/position 3.

| | | |
|---|---|---|
| 22<br>1 | 21<br>2 | 13<br>3 |
| 11<br>4 | 23<br>5 | <br>6 |
| 12<br>7 | 24<br>8 | <br>9 |

The second player will win even if the first player marks in cell/position 9 instead of in cell/position 8.

| | | |
|---|---|---|
| 22<br>1 | 21<br>2 | 13<br>3 |
| 12<br>4 | 23<br>5 | <br>6 |
| 11<br>7 | 14<br>8 | 24<br>9 |

# Game: Get-Away©

## Experiment Number: 1

Is it possible to *improve our ability to win* as we play more games? If yes, how?

Two Players play this game. Each Player has ten cards. Each card has a unique number on one of its side. The number is chosen from a range of numbers 0 to 9. The first Player chooses a card as per his/her wish and puts this card on the table such that the number on it is not disclosed. Next, the second Player picks a card of his/her choice and puts the card in the same way on the table. Now, the two Players disclose the numbers on their cards. They calculate the absolute difference of these two numbers. If the absolute difference is less than or equal to 2, then the first Player scores a point and pick next card of his choice from the cards he/she has and continue to play. Else, there is no score for any player but the game moves on to the next Player to pick a card of his/her choice and continue. A Match comprises of two players placing their cards in sequence and declaring the match point. A set of Matches makes a Game. Both players maintain their cumulative match points by summing the points they acquired throughout the Game. The Player who scores more cumulative match points of the two wins the Game.

Player A: Name: Roll No.:

| Number on the Card | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Chosen Card No. | | | | | | | | | | |

Player B: Name: Roll No.:

| Number on the Card | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Chosen Card No. | | | | | | | | | | |

Game No.: Threshold Number, n: 2

| Match No. | First Player | | Second Player | | Absolute Difference of Numbers, D = \|a - b\| | $D \le n$ $D \le 2$ | | Player Name that scored the Point | Match Point for Player A | Match Point for Player B |
|---|---|---|---|---|---|---|---|---|---|---|
| | Name | Number a | Name | Number b | | T | F | | | |
| 1 | | | | | | | | | | |
| 2 | | | | | | | | | | |
| 3 | | | | | | | | | | |
| 4 | | | | | | | | | | |
| 5 | | | | | | | | | | |
| 6 | | | | | | | | | | |
| 7 | | | | | | | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |
| 10 | | | | | | | | | | |
| Total Match Points | | | | | | | | | | |

Types of Behavior (Strategy and Heuristic) of Players:

1. Both Players play uniformly randomly (Unresponsive to opponent's actions) – Experiment 1
2. Player A always plays a number that is very close or the same as the last number disclosed by Player B as long as it is possible – Expt 2

Assumption: the players do not change the strategies during the game.

## Game: Get-Away©

### Experiment Number: 1.1

Is it possible to *improve our ability to win* as we play more games? If yes, how?

Two Players play this game. Each Player has Three cards. Each card has a unique number on one of its side. The number is chosen from a set of numbers {1,2,3}. The first Player chooses a card as per his/her wish and puts this card on the table such that the number on it is not disclosed. Next, the second Player picks a card of his/her choice and puts the card in the same way on the table. Now, the two Players disclose the numbers on their cards. They calculate the absolute difference of these two numbers. If the absolute difference is equal to 0, then the first Player scores a point and pick next card of his choice from the cards he/she has and continue to play. Else, there is no score for any player but the game moves on to the next Player to pick a card of his/her choice and continue. A Match comprises of two players placing their cards in sequence and declaring the match point. A set of Matches makes a Game. Both players maintain their cumulative match points by summing the points they acquired throughout the Game. The Player who scores more cumulative match points of the two wins the Game.

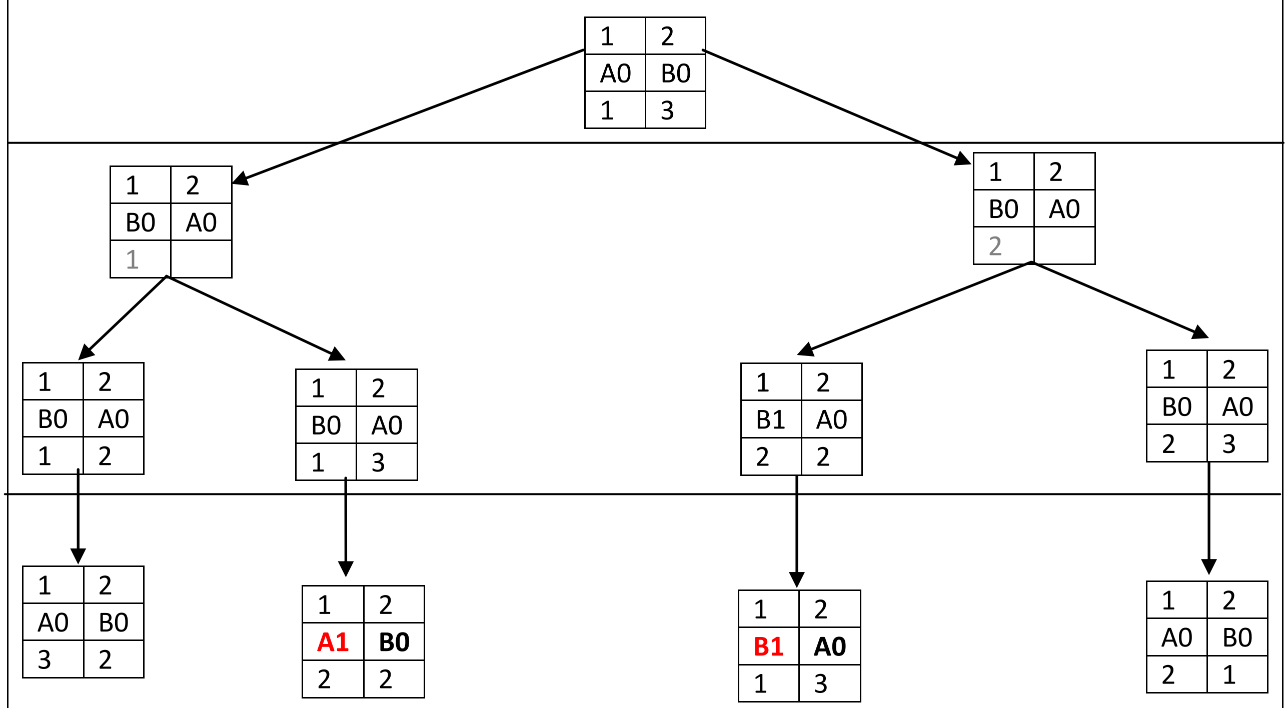



Figure 1. Part of the Game Tree for the Problem Get-Away with Numbers from Set = {1,2,3} and Threshold Number n equal to Zero.

Game: Get-Away©

Experiment Number: 1.1
Is it possible to *improve our ability to win* as we play more games? If yes, how?

Two Players play this game. Each Player has Three cards. Each card has a unique number on one of its side. The number is chosen from a set of numbers {1,2,3}. The first Player chooses a card as per his/her wish and puts this card on the table such that the number on it is not disclosed. Next, the second Player picks a card of his/her choice and puts the card in the same way on the table. Now, the two Players disclose the numbers on their cards. They calculate the absolute difference of these two numbers. If the absolute difference is equal to 0, then the first Player scores a point and pick next card of his choice from the cards he/she has and continue to play. Else, there is no score for any player but the game moves on to the next Player to pick a card of his/her choice and continue. A Match comprises of two players placing their cards in sequence and declaring the match point. A set of Matches makes a Game. Both players maintain their cumulative match points by summing the points they acquired throughout the Game. The Player who scores more cumulative match points of the two wins the Game.

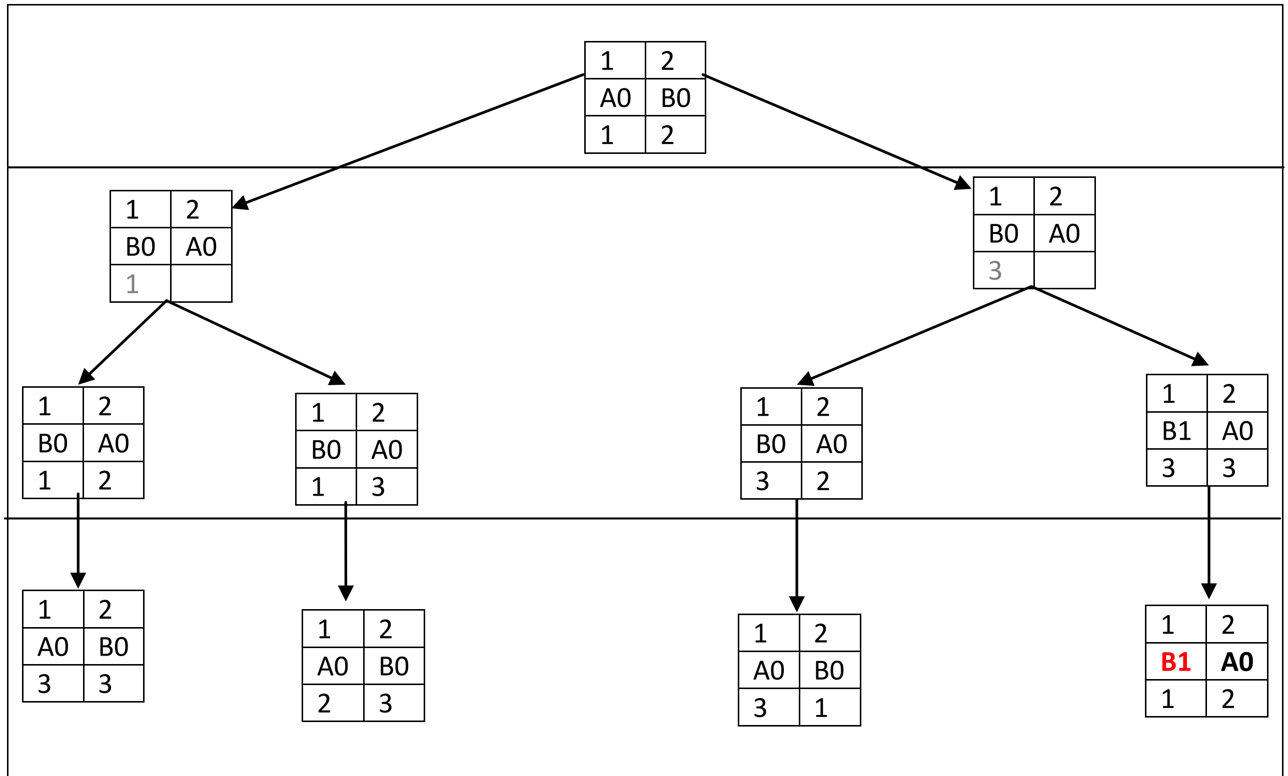



Figure 2. Part of the Game Tree for the Problem Get-Away with Numbers from Set = {1,2,3} and Threshold Number n equal to Zero.

Game: Get-Away©

Experiment Number: 1.1

Is it possible to *improve our ability to win* as we play more games? If yes, how?

Two Players play this game. Each Player has Three cards. Each card has a unique number on one of its side. The number is chosen from a set of numbers {1,2,3}. The first Player chooses a card as per his/her wish and puts this card on the table such that the number on it is not disclosed. Next, the second Player picks a card of his/her choice and puts the card in the same way on the table. Now, the two Players disclose the numbers on their cards. They calculate the absolute difference of these two numbers. If the absolute difference is equal to 0, then the first Player scores a point and pick next card of his choice from the cards he/she has and continue to play. Else, there is no score for any player but the game moves on to the next Player to pick a card of his/her choice and continue. A Match comprises of two players placing their cards in sequence and declaring the match point. A set of Matches makes a Game. Both players maintain their cumulative match points by summing the points they acquired throughout the Game. The Player who scores more cumulative match points of the two wins the Game.

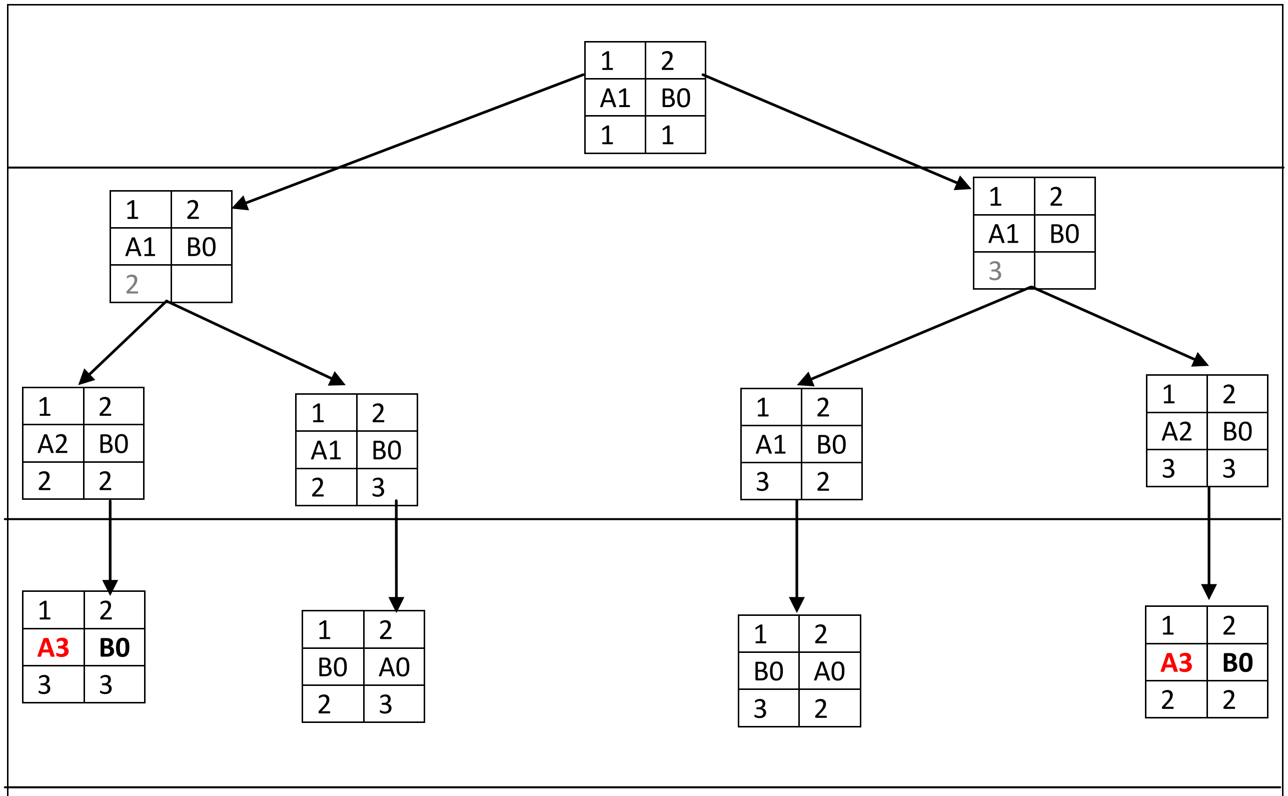



Figure 3. Part of the Game Tree for the Problem Get-Away with Numbers from Set = {1,2,3} and Threshold Number n equal to Zero.

Game Tree [see Complete Game Tree document for this problem given separately]

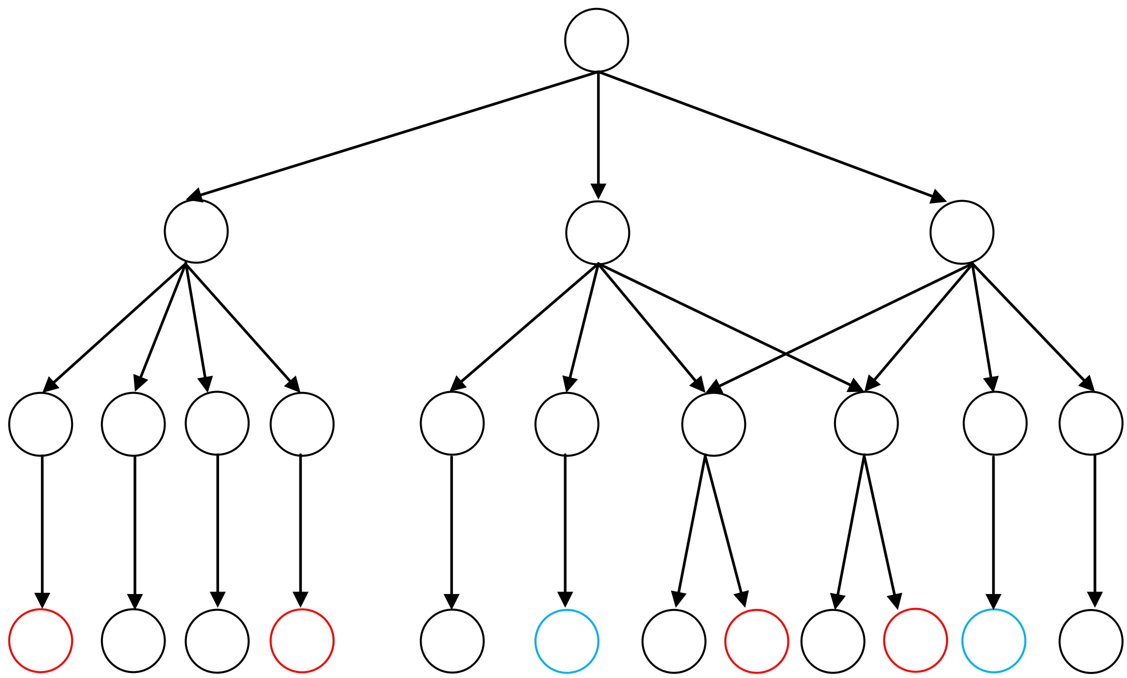


Figure 5. Complete Game Tree for the Problem Get-Away© with Numbers from Set = {1, 2, 3} and Threshold Number n equal to Zero when Player A plays first. A wins in {(A3, B0), (A3, B0), (A1, B0), (A1, B0)}. B wins in {(B1, A0), (B1, A0)} and the Game ends in a draw 6 times. For details of the nodes, see Figure 4.

Game: Lake Diggers© – Team A and Team B

An arbitrary shaped Area comprising of squares of unit size has to be dug into a lake. There are n number of diggers to do this job in a given number of units of time t. Where n is less than the number of squares in the arbitrary shaped area. Both teams A and B dig separate lakes of identical shapes in plan view to start with. Each digger can dig one cubic unit of volume in a unit time. Each digger can climb a step of only one unit and no more while doing the job. Which-ever team digs a more voluminous lake wins.

What is the maximum volume of the lake that can be dug? And what is the corresponding maximum depth (z) of the lake and its coordinates (x, y).

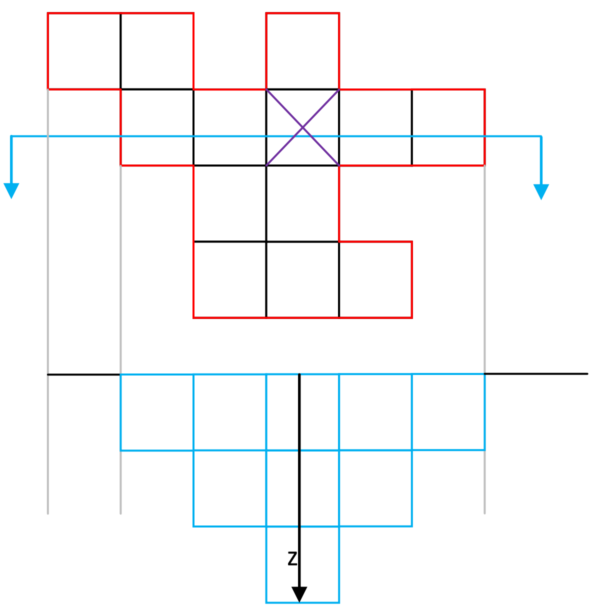



Figure 6. Example Lake with its plan view of arbitrary shaped area in red line and its corresponding sectional elevation in blue line.

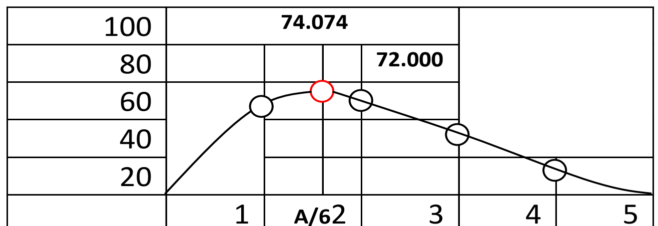
# Search for Optimal Value (Maxima) in Discrete and Continuous Domains

**Problem**: A tray can be made from a square lamina by removing square pieces of same size at all corners of the lamina and then folding the sides upright. If the size of the square lamina is 10 units x 10 units, and the corner square pieces can take any size from 1 unit x 1 unit onwards. Find the maximum volume of the tray, and the dimensions of the tray with maximum volume. Also if the corner squares can assume continuous values from 0 x 0 onwards, Find the ideal maximum volume of the tray possible and the corresponding dimensions of the tray.

**Solution**: Let A = 10 be the side of the square lamina. Let c be the side of the square being removed at all corners of the square lamina. Then the sides of the bottom of the tray will be a = A – 2c = 10 – 2(c = 1, 2, …), while the height of the tray is c.

1. For various discrete values of c, discrete volume V is calculated:

| c | a | $a^2$ | $V = a^2c$ |
|---|---|---|---|
| 0 | 10 | 100 | 0 |
| 1 | 8 | 64 | 64 |
| **2** | **6** | **36** | **72** |
| 3 | 4 | 16 | 48 |
| 4 | 2 | 4 | 16 |
| 5 | 0 | 0 | 0 |



Plot Showing Continuous and Discrete Values of Volume for Maxima.

2. General equation of volume in continuous domain is $V = (A - 2c)^2c = A^2c - 4Ac^2 + 4c^3$.
   The maxima is found by differentiating the above equation w.r.t c and equating it to zero.
   $dV/dc = A^2 - 8Ac + 12c^2 = 0$.
   Solving, we get $c = A/6$
   Substituting this in V, we get $V = [16/(36x6)]A^3$
   For A = 10, **V = 74.074**

| Algorithmic Time-Complexity Analysis<br>**For various sizes of given square lamina** | |
|---|---|
| Solution Method 1 (Discrete) | Solution Method 2 (Continuous) |
| For a square with side n, Total computational units involved = (n/2 volume computations – 1) + ((n/2-1) -1) volume comparisons for finding maximum value. T(n) = (n/2 – 1) + [(n/2 – 1) -1]. | For a square with side n, Total computational units involved = T(n) = 1 (constant). |