Jahangirnagar University

Department/Institute: Institute of Information Technology

Honours 4th **year Final Examination-20**20

Assignment for Final Examination

Course No.# IT- 4101

Course Title# Artificial Intelligences and Neural Networks

Name of the Student: Tahmina Harun Faria

Class Roll No. #	1961
Examination Roll No. #	172457
Registration No. #	45717
Academic Session #	2016 - 2017

Total number of written pages in the assignment # 113

Date of Submission: 24 August **, 2021**

Instructions:

1. Don't copy from other's assignment. Copying from others will be punished severely.
2. The student must submit the assignment **online (Google classroom/email/google form etc.)** as the course-teacher prescribes.
3. You must use **your name# your EXAM ID** only for naming your submitted file.

<u>Content</u>

(1)

Artificial Intelligence:

Artificial intelligence is the branch of computer science concerned with making computers behave like humans. Basically it is the science and engineering of making intelligent machines and intelligent computer programs.

The definitions of AI are categorized into four approaches. These are explained below:

a) Acting humanly:

To determine the behave of acting humanly, a test was proposed by Alan Turing in 1950. The computer passes the test if a human can not differentiate between a person's or a computer's response to some written questions. The computer requires the following capabilities to pass the test:

- Natural language processing - enable it to communicate in humanly language
- Knowledge representation - able to store what it knows or hears.
- Automated reasoning - able to use stored information in newer dimensions
- Machine learning - to adapt to new circumstances and detect and extrapolate patterns.

b) Thinking humanly:

Actual working of human mind needs to be taught –

- through introspection
- through psychological experiments

The interdisciplinary field of cognitive science brings together computer models from AI and experimental techniques from psychology to try to construct precise and testable theories of the working of the human mind.

c) Thinking rationally:

Thinking from a right and logical point of view is irrefutable reasoning process. It provides pattern for argument structures that always yielded correct conclusions when given correct premises. It created a new field called logic.

d) Acting rationally:

An agent is something that acts. They possess the following attributes:

- operating under autonomous control
- perceiving their environment
- persisting over a prolonged time period
- adapting to change.

(2) AI can do following tasks today:

- Autonomous planning and scheduling-

Remote agent generated plans from high-level goals specified from the ground. It can monitor the operation of the spacecraft as the plans were executed - detecting, diagnosing and recovering from problems as they occurred.

Example - NASA's remote agent program

- Game playing -

IBM's deep blue became the first computer program to defeat the world champion in a chess match when it bested Gary Kasparov by a score of 3.5 to 2.5 in an exhibition match.

Autonomous Control:

The ALVINN computer vision system was trained to steer a car to keep it following a lane. It was placed in CMU's NAVLAB computer-controlled minivan and used to navigate accross the United States for 2850 miles it was in control of steering the vehicle 98% of the time.

Diagnosis:

Medical diagnosis programs based on probabilistic analysis have been able to perform at the level of an expert physician in several areas of medicine.

Logistics Planning:

During the Persian gulf crisis of 1991, U.S. forces deployed a dynamic analysis and replanning tool, DART, to do automated logistics planning and scheduling for transportation. This involved upto 50,000 vehicles, cargo and people at a time and had to account for starting points, destinations, routes and conflict resolution among all parameters. The AI planning techniques allowed a plan to be generated in hours that would have taken weeks with older methods.

Robotics:

Many surgeons now use Robot assistants in microsurgery. HipNav is a system that uses computer vision techniques to create a three dimensional model of a patient's internal anatomy and then uses robotic control to guide the insertion of a hip replacement prosthesis.

⊞ Language understanding and problem solving.

PROVERB is a computer program that solves crossword puzzles better than most humans, using constraints on possible word fillers, a large database of past puzzles and a variety of information sources including dictionaries and online databases such as a list of movies and the actors that appear in them.

3

Agent:

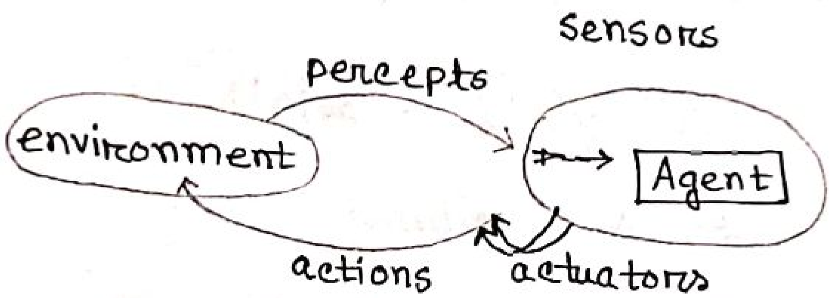
An agent is anything that can be viewed as perceiving its environment through sensors and sensors acting upon that environment through actuators.

Agent Function:

Agent function is the mathematical expression of an agent's behavior that maps any given percept sequence to an action.

(4)

The agents interact with environments through sensors and actuators in the following manner:



Vacaum cleaner example -

This particular world has two locations: squares A and B. The vacuum agent perceives which square it is in and whether there is dirt in the square. It can choose to move left, move right, suck up the dirt or do nothing. One very simple agent function is the following: if the current square is dirty, then suck, otherwise move to the other square.

Agent program –

function REFLEX – VACUUM – AGENT (location, status)

if status = Dirty then return Suck

else if location = A then return Right

else if location = B then return left

5. The PEAS description are shown below:

a) Automated Taxi –

* Performance measure –

Safe, fast, legal, comfortable trip, maximize profit.

* Environments –

Roads, other traffic, pedestrians, customers.

Actuators -

steering, accelator, brake, signal, horn, display.

Sensors -

Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboards, accelometer.

b) Medical diagnosis system -

Performance measure -

Healthy patient, minimize costs, lawsuits.

Environment -

Patient, hospital, staffs.

Actuators -

Display questions, tests, diagnosis, treatments, referrals.

Sensors -

Keyboard entry of symptoms, findings, patient's answers.

c) Part - picking robot -

Performance measure-

Percentaage of parts in correct bins

Environment-

Conveyer belt with parts, bins.

Actuations -

Jointed arm and hand

Sensors-

Camera, joint angle sensors.

d) Refinery controller:

Performance measure-

maximize purity, yield, safety.

Enviroment-

refinery, operator

Actuations -

Valves, pumps, heater, display.

Sensors -

Temperature, pressure, chemical sensors

5

(6) The properties of task environments are explained below :

⊞ Fully observable vs. partially observable :

If an agent's sensors give it access to the complete state of the environment at each point in time, then the task environment is fully observable. A task environment is effectively fully observable if the sensors detect all aspects that are relevant to the choice of action. An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data.

⊞ Deterministic vs. stochastic :

If the next state of the environment is completely determined by the current state and the action executed by the agent, then the environment is deterministic, otherwise it is stochastic.

Episodic vs. sequential –

In an episodic task environment, the agent's experience is divided into atomic episodes. Each episode consists of the agent perceiving and then performing a single action.

In sequential environments, the current decision could affect all future decisions.

Discrete vs. continuous –

The discrete/continuous distinction can be applied to the state of the environment, to the way time is handled and to the percepts and actions of the agent. For example, a discrete-state environment such as a chess game has a finite numbers of distinct states. Chess also has a discrete set of percepts and actions.

Single agent vs. multiagent -

An agent solving a crossword puzzle by itself is clearly in a single agent environment, whereas an agent playing chess is in a two agent environment.

The hardest case is partially observable, stochastic, sequential, dynamic, continuous and multiagent.

⑦ Properties of the following environments -

Task environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
a) Taxi driving	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi
b) Medical Diagnosis System	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
c) Part picking robot	Partially	Stochastic	Episodic	Dynamic	Continuous	Single
d) Refinery Controller	Partially	Stochastic	Sequential	Dynamic	Continuous	Single

(8) The difference between agent program and agent function is that the agent program takes the current percept as input whereas the agent function takes the entire percept history.

(9) The drawbacks of table driven agent are :

a) Table lookup of percept-action pairs defining all possible condition-action rules necessary to interact in an environment.

b) Too big to generate and to store. For example, chess has about 10^{120} states.

c) No knowledge of non-perceptual parts of the current state.

d) Not adaptive to changes in the environment. It requires entire table to be updated if changes occur.

e) Looping : can not make actions conditional.

f) take a long time to build the table.

g) No autonomy

h) Even with learning need a long time to learn the table entries.

(10) There are five types of agents. These are explained below:

a) Table - driven agents -
use a percept sequence / action table in memory to find the next

action. They are implemented by a large lookup table.

b) Simple reflex agents -

are based on condition-action rules, implemented with an appropriate production system. They are stateless devices which do not have memory of past world states.

c) Agents with memory-

have internal state which is used to keep track of past states of the world

d) Agents with goal-

are agents that, in addition to state information, have goal information that describes desirable situations. Agents of this kind take future events into consideration.

e) Utility based agents -

base their decisions on classic axiomatic (evident without proof or argument) utility theory in order to act rationally.

11

a) Table-driven agents :

It uses a percept sequence / action table in memory to find the next action. They are implemented by a large lookup table. The table-driven agent program is invoked for each new percept and returns an action each time. It has a table lookup of percept-action pairs defining all possible condition-action rules necessary to interact in an enviroment. It does not have any knowledge of non-perceptual & part of the current state.

b) Simple reflex agents-

The simplest kind of agent is the simple reflex agent. These agents select actions on the basis of the current percept, ignoring the rest of the percept history.

Characteristics:

- Only works if the environment is fully observable.
- Lacking history, easily get stuck in infinite loops.
- One solution is to randomize action.

c) Model based reflex agents -

The most effective way to handle partial observability is for the agent to keep track of the part of the world it cannot see now. The agent should maintain some sort of internal state that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state. Updating this internal state information as time goes by requires two kinds of knowledge to be encoded in the agent program. First some information about how the world evolves independently of the agent is needed. Then how the agent's own actions affect the world- that information is needed. This information about " how the world working - whether implemented in simple boolean circuits or in complete scientific theories is called a model of the world. An agent that uses such a MODEL-BASED model is called a model-based agent.

d) Goal based agent-

Knowing about the current state of the environment is not always enough to decide what to do. As well as a current state description, the agent needs some sort of goal information that describes situations that are desirable. The agent program can combine this with information about the results of possible actions (the same information as was used to update internal state in the reflex agent) in order to choose actions that achieve the goal.

e) Utility based agents –

Goals alone are not really enough to generate high-quality behavior in most environments. Goals just provide a crude binary distinction "happy" and "unhappy" states, whereas a more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent if they could be achieved. Because "happy" does not sound very scientific, the customary terminology is to say that if one world state is preferred to another, then it has higher utility for the agent.

12. A well defined problem can be described by following characteristics:

- Initial State
- Operator or successor function-

for any state x returns s(x), the set of states reachable from x with one action.

- State space-

all states reachable from initial by any sequence of actions

- Path-

sequence through state space.

- Path cost-

function that assigns a cost to a path. Cost of a path is the sum of the costs of individual actions along the path.

- Goal test –

  test to determine if at goal state

(13) A problem solving agent is a goal-based agent. It decides what to do by finding sequence of actions that lead to desirable states. The agent can adopt a goal and aim at satisfying it.

Goal formulation based on the current situation and the agent's performance measure is the first step in problem solving.

Problem formulation is the process of deciding what actions and states to consider given a goal.

(14) An agent design assumes its environment is, if it does not have any idea of it – are as follows:

- Static –

The entire process carried out without paying attention to changes that might be occuring in the environment.

- Observable –

The initial state is known and the agent's sensor detects all aspects that are relevant to the choise of action

- Discrete –

With respect to the state of the environment and percepts and actions so that alternate course of action can be taken.

• Deterministic :-

The next state of the environment is completely determined by the current state and the actions executed by the agent. Solutions to the problem are single sequence of actions.

15 A problem can be formally defined by four components :

- The initial state that the agent starts in.
- A successor function returns the possible actions available to the agent. Given a state x, SUCCESSOR-FN(x) returns a set of {action, successor} ordered pairs where each action is one ~~on~~ of the

legal actions in state x and each successor is a state that can be reached from x by applying the action.

- State space –

The set of all states reachable from initial state. The state space forms a graph in which the nodes are the states and the arcs between nodes are actions.

- A path in the state space is a sequence of states connected by a sequence of actions.

- The goal test determines whether the given state is a goal state.

(16) a) 8-queens problem:

The goal of 8-queens problem is to place 8-queens on the chessboard such that no queen attacks any other. { A queen attacks any piece in the same row, column or diagonal.

- Initial state – No queen on the board.
- States – Any arrangement of 0 to 8 queens on board is a state.
- Successor function – Add a queen to any empty square.
- Goal test – 8 queens are on the board, none attacked.

b) Airline travel problem:

The airline travel problem specifies as follows -

- States - each state is represented by a location and current time.
- Initial state - this is specified by the problem.
- Successor function - this returns the states resulting from taking any scheduled flight, leaving later than the current time plus the within-airport transit time, from the current airport to another.
- Goal test - passengers reaching at the destination by some prespecified time.
- Path cost - This depends upon the monetary cost, waiting time, flight time, customs and immigration procedures, seat quality, type of air plane, time and so on.

c) The 8-puzzle problem -

An 8-puzzle consists of a 3X3 board with eight numbered tiles and a blank space. A tile adjacent to the blank space can slide into the space.

- 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. It can be noted that any given goal can be reached from exactly half of the possible initial states.

- Successor function - This generates the legal states that result from trying the four actions (blank moves left, right, up or down)

• Goal test - This checks whether the state matches the goal configuration of the blocks in an ascending sequence.

• Path cost - Each step costs 1, so the path cost is the number of steps in the path.

17 Node:

A node is a book keeping data structure used to represent the search tree.

Fringe:

Fringe is a collection of nodes that have been generated but not yet been expanded. Each element of the fringe is a leaf node with no successor in the tree.

A node is a data structure with five components –

- State – a state in the state space to which the node corresponds.
- Parent node – the node in the search tree that generated this node.
- Action – the action that was applied to the parent to generate the node.
- Path-cost – the cost denoted by $g(n)$ of the path from the initial states to the node, as indicated by the parent pointers.
- Depth – the number of steps along the path from the initial state.

(18) General tree search algorithm:

```
function TREE-SEARCH( problem, fringe)
returns a solution, or failure
fringe ← INSERT (MAKE-NODE (INITIAL-
            STATE [problem], fringe)
loop do
    if Empty? (fringe) then return failure
    node ← REMOVE-FIRST (fringe)
    if GOAL-TEST [problem] applied to
    STATE [node] succeeds
        then return SOLUTION(node)
    fringe ← INSERT-ALL(EXPAND (node,
            problem), fringe)
```

```
function EXPAND (node, problem) returns a
  set of nodes

    successors ← the empty set
    for each (action, result) In SUCCESSOR-FN
      [problem] (STATE [node]) do
          s ← a new NODE
          STATE [s] ← result
          PARENT-NODE [s] ← node
          ACTION [s] ← action
          PATH-COST [s] ← PATH-COST [node]
                      + STEP-COST(node,
                                  action, s)
          DEPTH [s] ← DEPTH [node] + 1
          add s to successors
    return successors.
```

(19) Five uninformed search strategies are -

a) Breadth first search

b) Uniform cost search

c) Depth first search

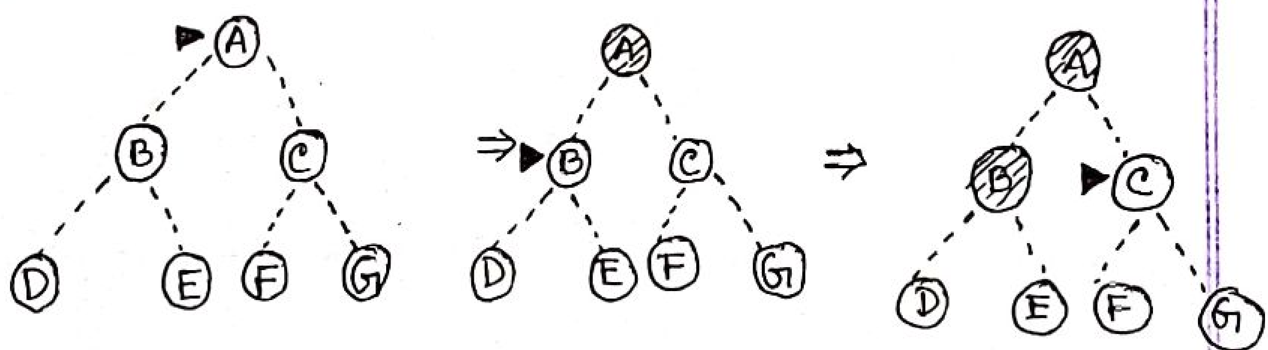
d) Depth limited search

e) Iterative deepening search

Breadth first search -

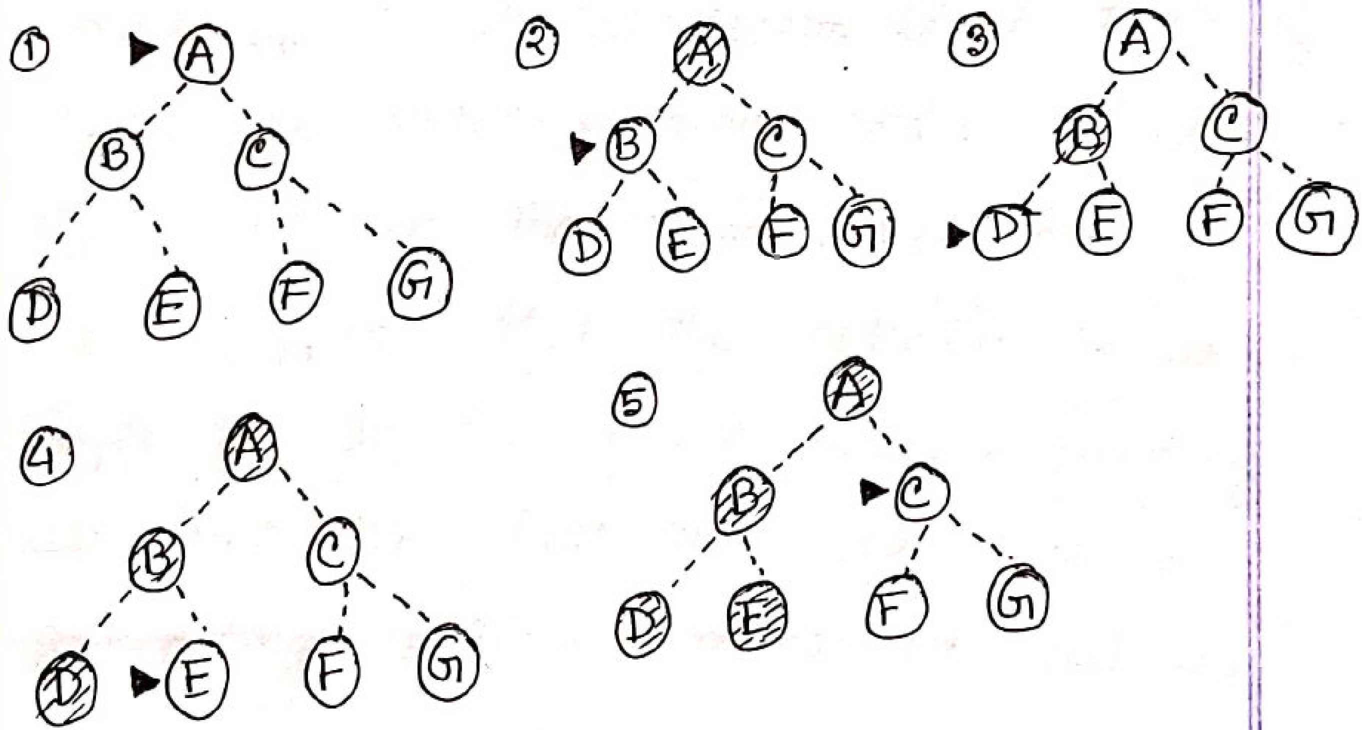
It is a simple strategy in which the root node is expanded first, then all successors of the root node are expanded next, then their successors and so on.

Example:



Depth First Search:

It always expands the deepest node in the current fringe of the search tree. The progress of the search proceeds immediately to the deepest level of the search tree where the nodes have no successor. Example -



(20) The time complexity for breadth-first search, assuming every state has p successors ~~b~~ :

The root of the search tree generates p nodes at the first level, each of which generates p more nodes, for a total of p^2 at the second level. Each of these generates p more nodes, yielding p^3 nodes at the third level and so on. Now suppose that the solution is at depth d. In the worst case, expanding all but the last node at level d, generating $p^{d+1} - p$ nodes at level $d+1$. Then the total number of nodes generated is :

$$p + p^2 + p^3 + \dots + p^d + (p^{d+1} + p) = O(p^{d+1})$$

Every node that is generated must remain in memory, because it is either part of the fringe or is an ancestor of a fringe node.

(21) The final state of vacuum problem if it is sensorless –

i) No sensor
ii) initial state {(1, 2, 3, 4, 5, 6, 7, 8)
iii) after action [Right] (2, 4, 6, 8)
iv) after action [Suck] (4, 8)
v) after action [left] (3, 7)
vi) after action (Suck) [7]

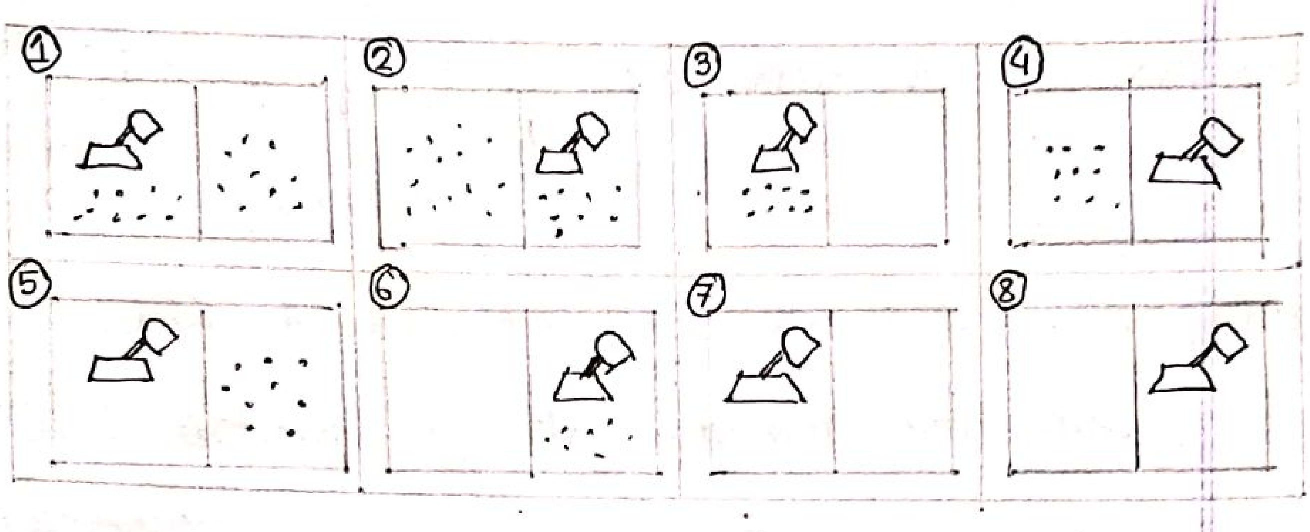


Figure - Initial state (1, 2, 3, 4, 5, 6, 7, 8)

Answer - [Right, suck, left, suck]

Belief state - such state that agent belief to be there.

(22) Following are three distinct problem types lead by partial information :

i) Sensorless problems - If the agent has no sensor at all

ii) Contingency problem- If the environment is partially observable or if actions are uncertain.

iii) Exploration problem- When the states and actions of the environment are unknown.

(23) Heuristic Search Strategy-

Heuristic or informed search strategy is one of the problem strategy that uses problem-specific knowledge beyond the definition of the problem itself. It can find solutions more efficiently than uninformed strategy.

A heuristic function or simply a heuristic is a function that ranks alternatives in various search algorithms at each branching up step based on an available information in order to make a decision which branch is to be followed during a search.

(24) <u>Greedy best-first search:</u>

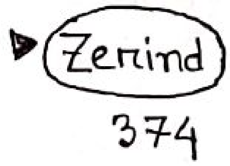
Greedy best-first search tries to expand the node that ~~i~~ is closest to the goal, on the grounds that is likely to a solution quickly. It evaluates the nodes by using heuristic function $f(n) = h(n)$.

Properties:

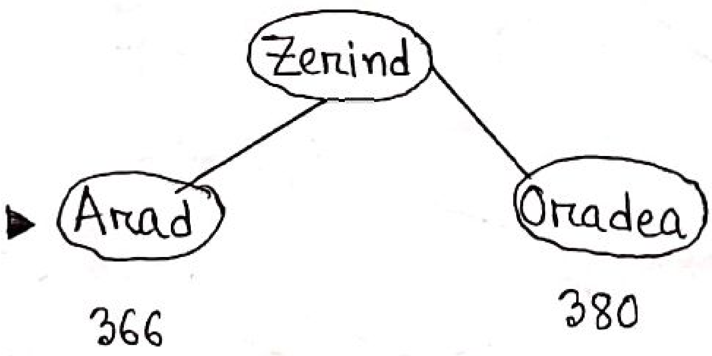
- Not complete, get stuck in loops.
- Time complexity – $O(bm)$
- Space complexity – $O(bm)$
- Not optimal.

(25) Path from Zerind to Bucharest using greedy best-first search

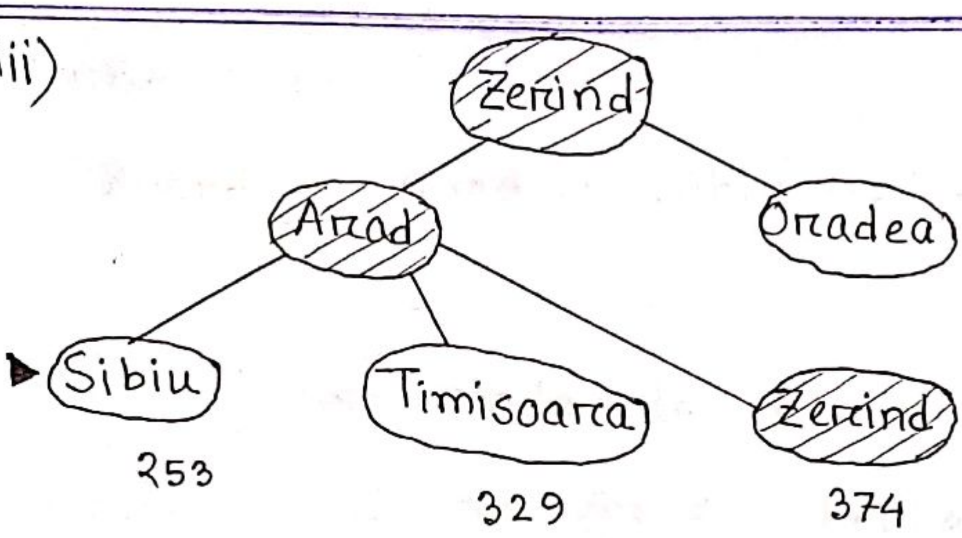
i)



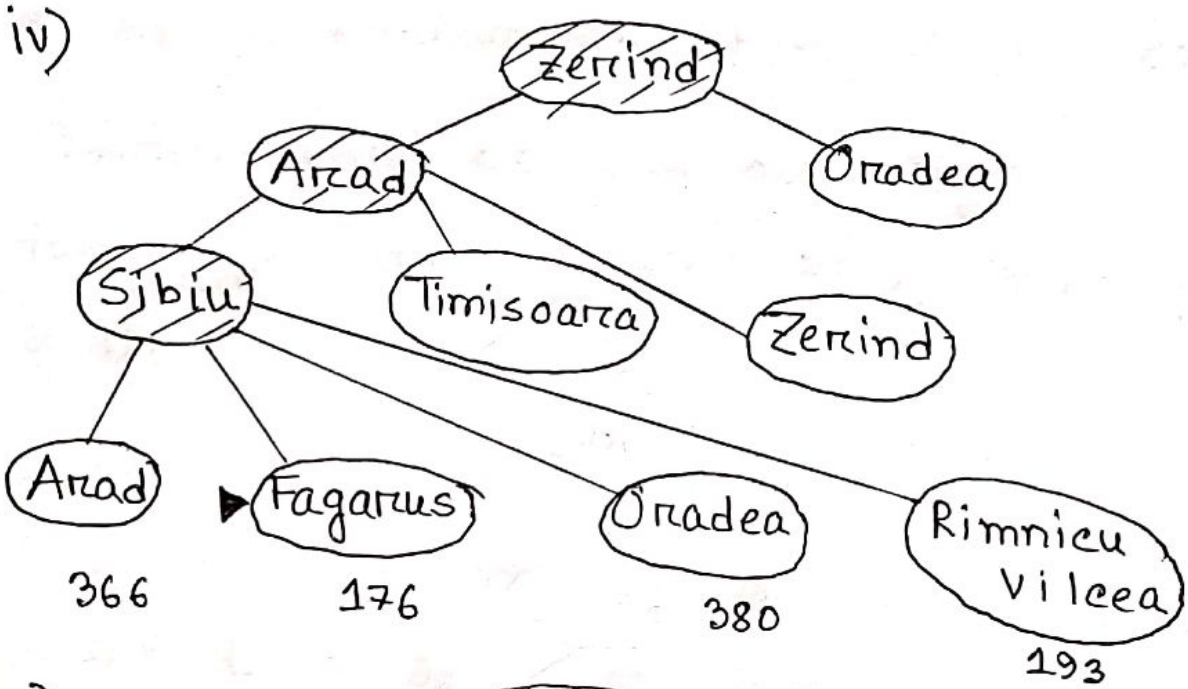
ii)



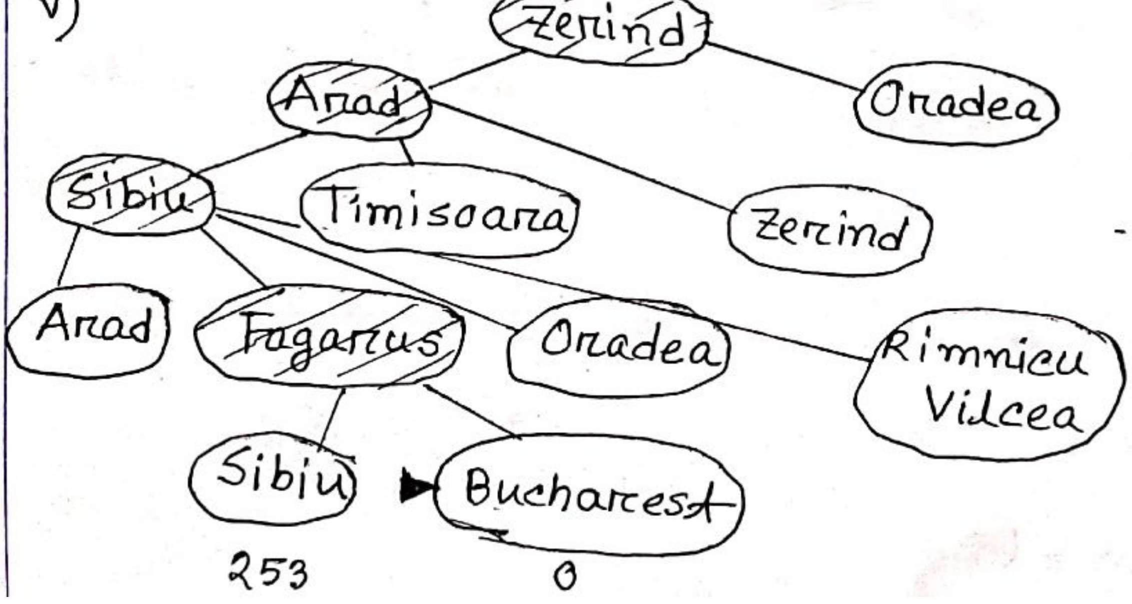
iii)


Zerind
Arad
Oradea
Sibiu
Timisoara
Zerind
253
329
374

iv)


Zerind
Arad
Oradea
Sibiu
Timisoara
Zerind
Arad
Fagarus
Oradea
Rimnicu Vilcea
366
176
380
193

v)


Zerind
Arad
Oradea
Sibiu
Timisoara
Zerind
Arad
Fagarus
Oradea
Rimnicu Vilcea
Sibiu
Bucharest
253
0

Path-

Zerind → Arad → Sibiu → Fagarus → Bucharest.

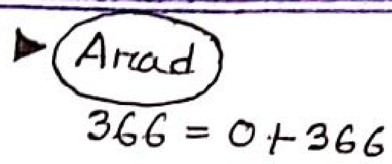
The first node to be expanded from Zerind will be Arad, because it is closer to Bucharest. Then Sibiu and Fagarus will be expanded. Fagarus in turn generates Bucharest, which is the goal.

(26) A^* Search is the most widely used form of best-first search. The evaluation function $f(n)$ is obtained by combining

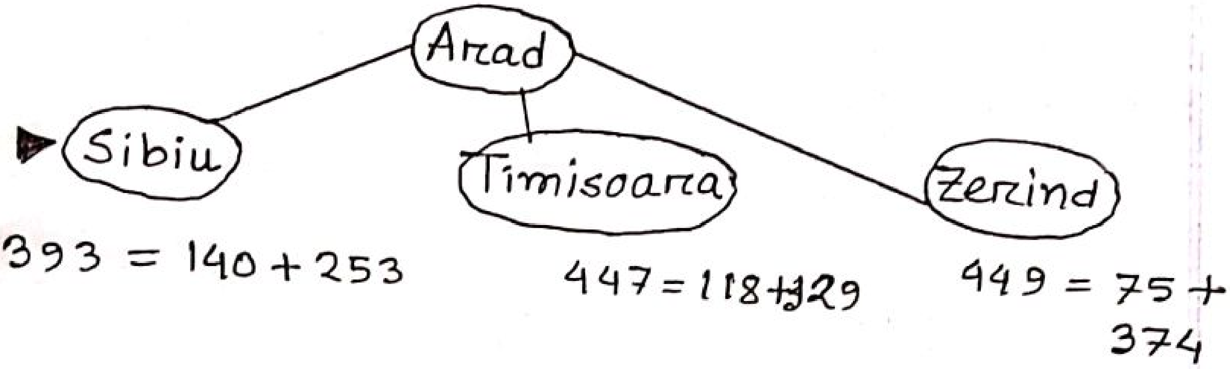
1) $g(n)$ = the cost to reach the node

2) $h(n)$ = the cost to get from the node to the goal: $f(n) = g(n) + h(n)$

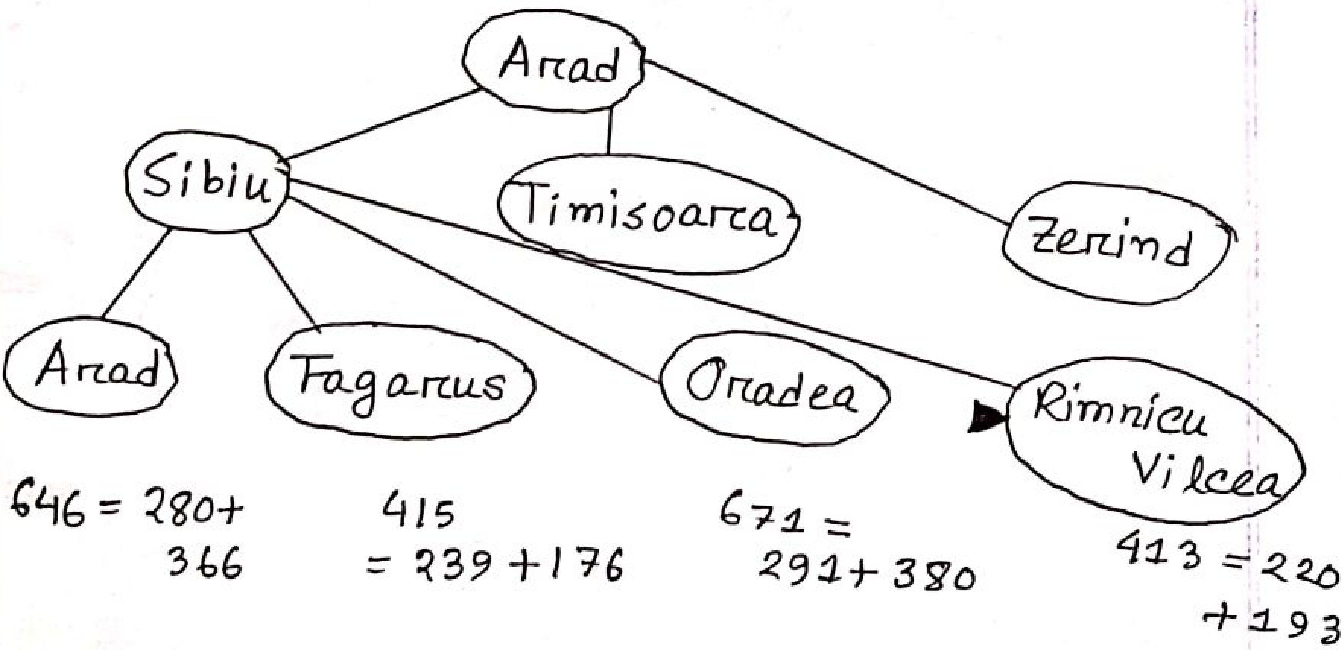
a) The initial state -



b) After expanding Arad -

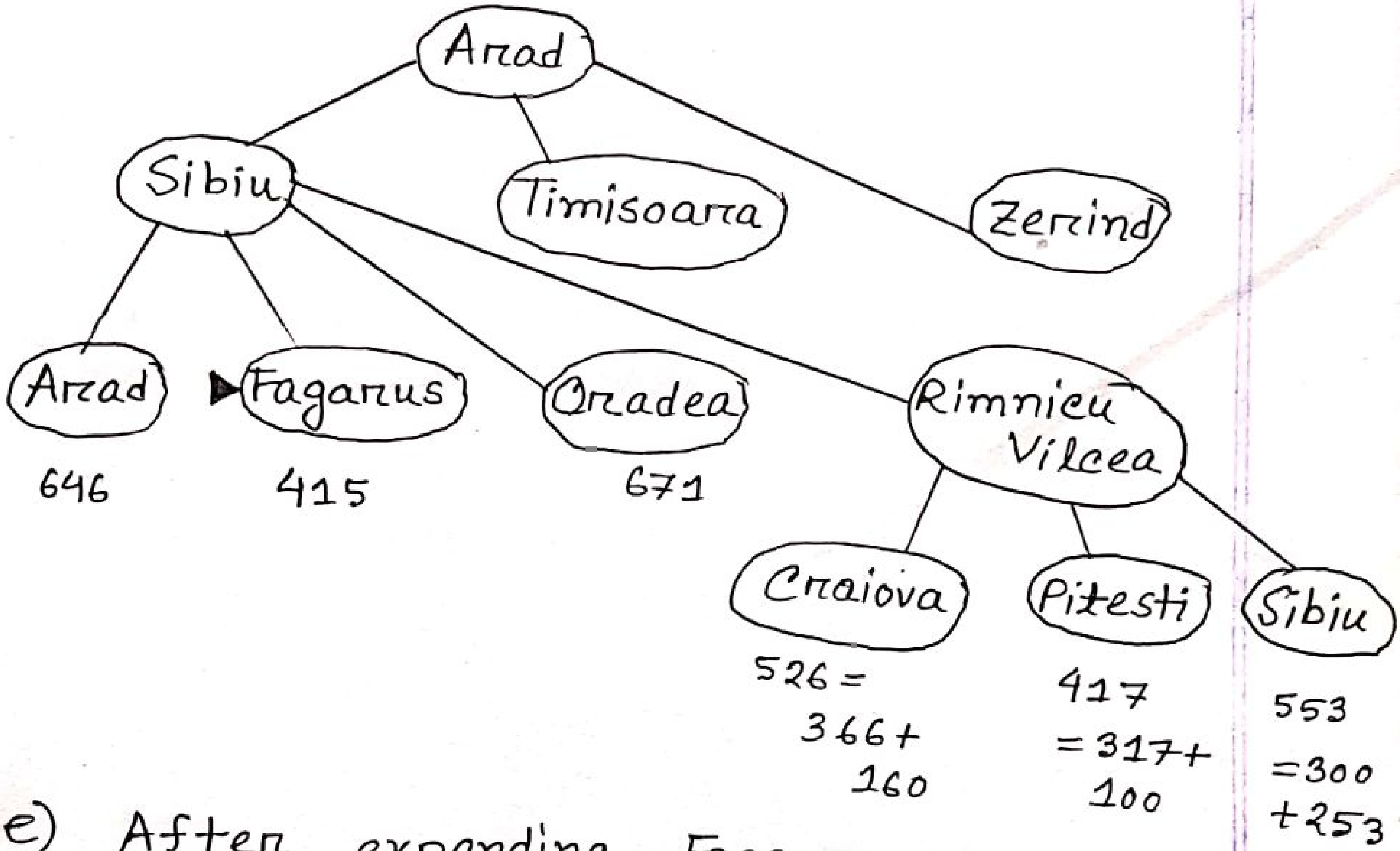


c) After expanding sibiu -

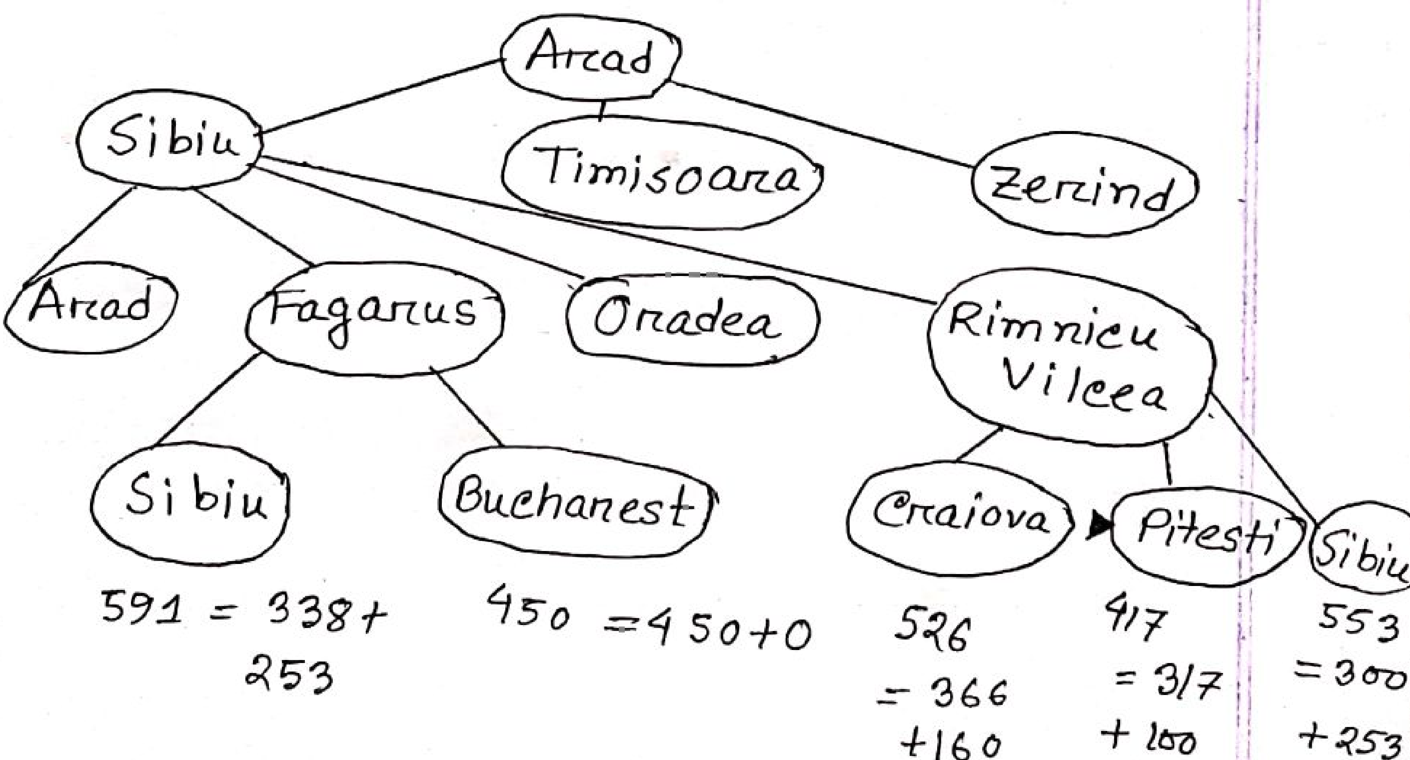


Scanned with CamScanner

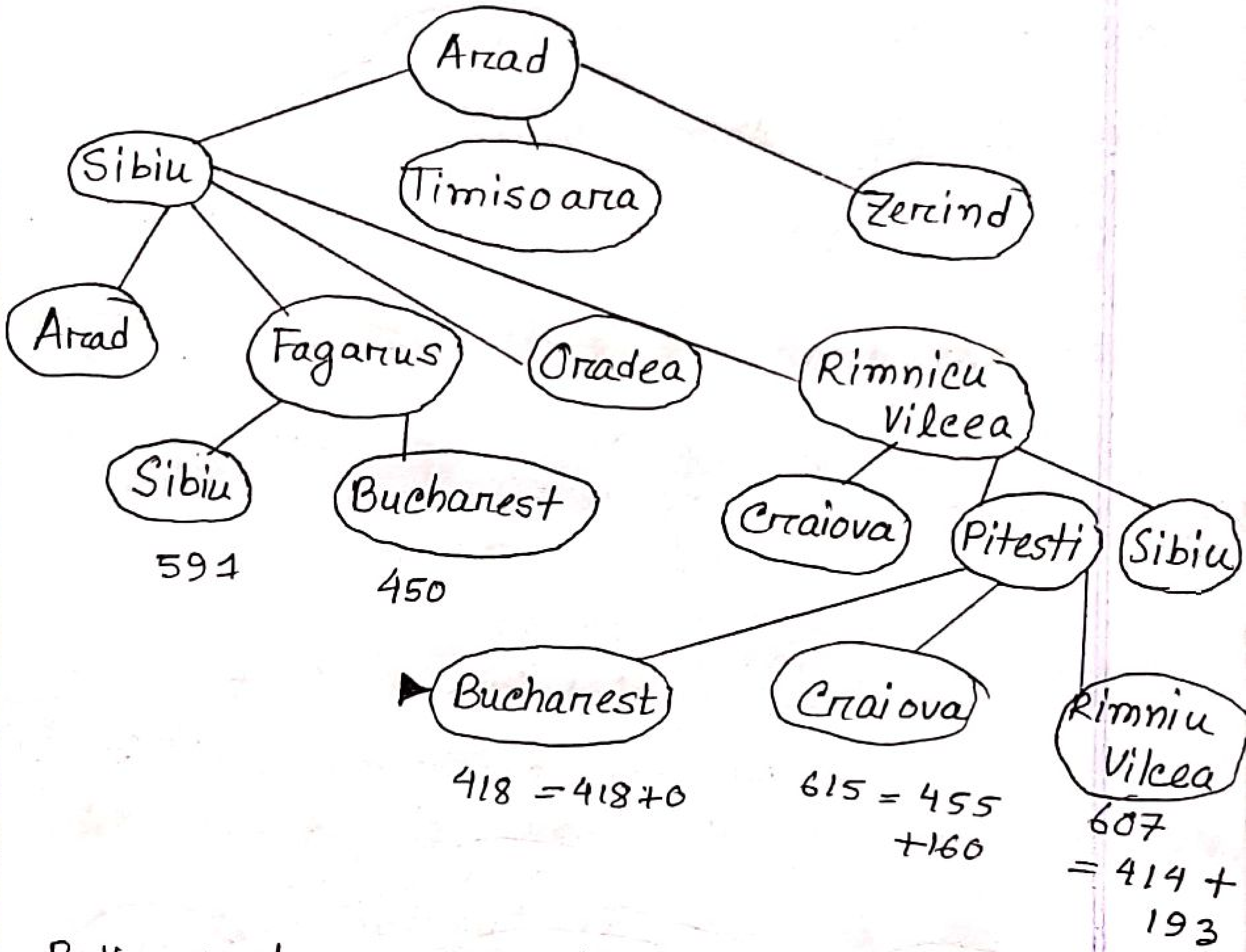
d) After expanding Rimnicu Vilcea -



e) After expanding Fagarus -



f) After Expanding Pitesti –



Path cost → 418 = 418+0.

27

A* search is the most widely used form of best-first search. The evaluation function $f(n)$ is obtained by combining -

1) $g(n)$ - the cost to reach the node
2) $h(n)$ - the cost to get from the node to the goal.

$$f(n) = g(n) + h(n)$$

A* search is both optimal and complete. A* is optimal if $h(n)$ is an admissable heuristic. The straight line distance h_{SLD} can not be an overestimate. That is $h(n)$ never overestimates the cost to reach the goal.

(28)

The minimax algorithm-

This algorithm computes the minimax decision from the current state. It uses a simple recursive computation of the minimax values of each successor state, directly implementing the defining equations. The recursion proceeds all the way down to the leaves of the tree and then the minimax values are backed up through the tree as the recursion unwinds. The minimax algorithm performs a complete depth first exploration of the game tree. If the maximum depth of the tree is m and there are b legal moves at each point, then the time complexity is $O(b^m)$ and space complexity is $O(bm)$

Alpha - Beta Pruning -

The problem with minimax search is that the number of game states it has to examine is exponential in the number of moves. But the exponent can not be eliminated but it can be effectively cut in half. The technique is called alpha beta pruning, when applied to a minimax tree, it returns the same move as minimax would, but prunes away branches that can not possibly influence the final decision.

- α : the value of the best choice at any choice point along the path of MAX.

- β : the value of best choice at any choice point along the path of MIN.

Alpha Beta search updates the values of α and β as it goes along and prunes the remaining branches at a node as soon as the value of current node is known to be worse than the current α and β value for MAX and MIN, respectively.

29

The cells that perform information processing in the brain is called neuron. It is the fundamental functional unit of all nervous system tissue. A neuron may connect to as many as 100,000 other neurons.

The information transmission happens at the synapses. The spikes travelling along the axon of the presynaptic neuron trigger the release of neurotransmitter substances at the synapse. The neurotransmitters cause excitation or inhibition in the dendrite of the post synaptic neuron. The integration of the excitatory and inhibitory signals may produce spikes in the post synaptic neuron. That is how brain works and information is transmitted.

30 Real and artificial neuron:

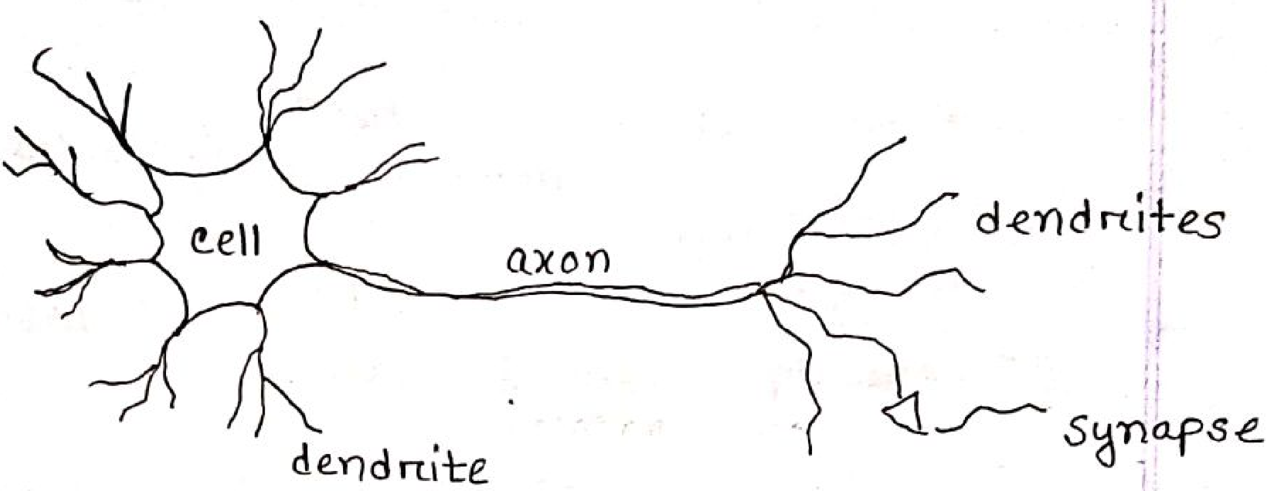
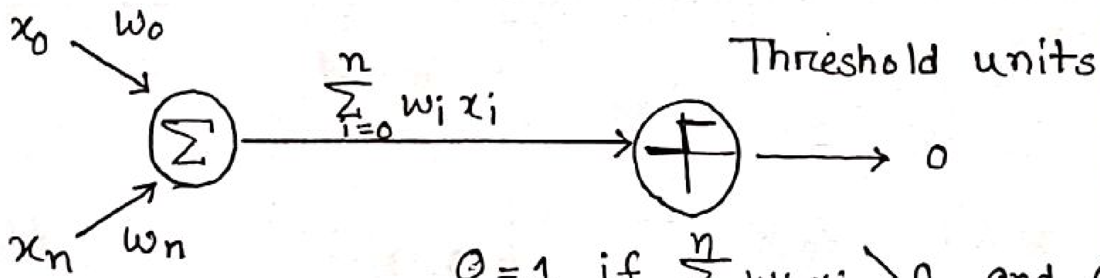


Figure- Real Neuron



$O = 1$ if $\sum_{i=0}^{n} w_i x_i > 0$ and 0 o/w

Differences:

Parameters	Real Neuron	Artifical Neuron
Speed	Slower in processing infor-mation	Fast in processing
Processing	Parallel Operation	Sequential mode
Size	Big size	Small size
Complexity	More	Less
Fault Tolerance	Tolerant	Not fault tolerant
Control Mechanism	No central control for processing information	Control unit monitors all the activities

31

a) Single perceptron-

It is a binary classifier which can decide whether or not an input, represented by a vector of numbers. It can learn any linearly separable function. It is a single unit network. The weight is changed by an amount proportional to the difference between the desired output and actual output.

b) Linear neurons -

Linear neurons are quite popular and useful for applications such as interpolation. However, they have a serious limitation - each neuron computes a linear function and therefore the overall network function $f: R^m \rightarrow R^n$ is also linear. This means that if an input vector x results in an output vector y, then for any factor ϕ the input $\phi \cdot x$ will result in the output $\phi \cdot y$.

c) Gaussian neuron -

This type of neurons are able to realize non-linear functions. Therefore, networks of Gaussian units are in principle unrestricted with regard to the functions that they can realize. The drawback of gaussian neurons is that it has to be made sure that their net input does not exceed 1.

d) Sigmoidal neuron-

Sigmoidal neurons accept any vectors of real numbers as input and they output a real number between 0 and 1.

§ These are the most common types of artifical neuron in learning networks. A network of sigmoidal units with m input neurons and n output neurons realizes a network function

$$f : R^m \rightarrow (0,1)^m$$

(32)

Supervised Learning -

A supervised learning algorithm learns from labeled training data that helps to predict outcomes for unforseen data. It means some data is already tagged with the correct answer.

Unsupervised Learning -

Unsupervised learning is a machine learning technique where the model does not need to supervise the model. It mainly deals with the unlabelled data. It finds all kind of unknown patterns in data.

Differences:

Parameters	Supervised Learning	Unsupervised Learning
1) Process	input and outputs are given	Only inputs are given
2) Input data	Labelled data	Unlabelled data
3) Complexity	Simpler method	Complex
4) Accuracy	Highly accurate	Less accurate
5) Real time Learning	Not possible	Possible
6) Number of Classes	Known	Unknown
7) Algorithms used	Support vector machine, neural network, Linear and logistics regression, random forest and classification trees	Cluster Algorithms, K-means, hierarchical clustering etc.

(33) Back propagation algorithm for updating weights in a multilayer network :

(1) Initialize the weights in the network (often randomly)

(2) repeat

for each example e in the training set do

i) O = neural-net-output (network, e) ; forward pass

ii) T = teacher output for e.

iii) Calculate error (T-O) at the output units

iv) Compute $w_j = w_j + \alpha * Err * I_j$ for all weights from hidden layer to output layer; backward pass.

v) Compute $w_j = w_j + \alpha * Err * I_j$ for all weights from input layer to hidden layer ; backward pass continued.

vi vi) Update the weights in the network.

end

(3) until all examples classified correctly or stopping criterion met.

(4) return (network)

(34) Neural Networks :

Advantages -

- Distributed computations
- Simple computations
- Robust with respect to noisy data
- Robust with respect to node failure.
- Empirically shown to work well for many problem domains
- Parallel processing

Disadvantages -

- Training is slow
- Interoperability is hard
- Network topology layouts ad hoc.
- Can be hard to debug.
- May converge to a local, not global, minimum of error.
- May be hard to describe a problem in terms of features with numerical values.

35 Steps of training in a back propagation network –

i) Initialize weights with small, random values.

ii) While stopping condition is not true.

– for each training pair (input/output):

• each input unit broadcasts its value to all hidden units.

• each hidden unit sums its input signals and applies activation function to compute its output signal

• each hidden unit sends its signal to the output units.

• each output unit sums its input signals and applies its activation function to compute its output signal.

iii) Each output computes its error term, its own weight correction term and its bias (threshold) correction term and sends it to layer below.

iv) Each hidden unit sums its delta inputs from above and multiplies by the derivative of its activation function; it also computes its own weight correction term and its bias correction term.

v) Each output unit updates its weights and bias.

vi) Each hidden unit updates its weights and bias.

36 a) Net input and output calculation -

Unit, j	Net input, I_j	Output, O_j
4	$0.2 + 0 - 0.5 - 0.4 \cancel{-0} = -0.7$	$1/(1+e^{0.7}) = 0.332$
5	$-0.3 + 0 + 0.2 + 0.2 = 0.1$	$1/(1+e^{-0.1}) = 0.525$
6	$(-0.3)(0.332) - (0.2) * (0.525) + 0.1 = -0.105$	$1/(1+e^{0.105}) = 0.474$

b) calculation of error at each node -

Unit, j	Error$_j$
6	$(0.474)(1-0.474)(1-0.474) = 0.1311$
5	$(0.525)(1-0.525)(0.1311)(-0.2) = -0.0065$
4	$(0.332)(1-0.332)(0.1311)(-0.3) = -0.0087$

c) Calculation for weight updating:

Weight	New value
w_{46}	$-0.3 + (0.9)(0.1311)(0.332) = -0.261$
w_{56}	$-0.2 + (0.9)(0.1311)(0.525) = \cancel{0} -0.138$
w_{14}	$0.2 + (0.9)(-0.0087)(1) = 0.192$
w_{15}	$-0.3 + (0.9)(-0.0065)(1) = -0.306$
w_{24}	$0.4 + (0.9)(-0.0087)(0) = 0.4$
w_{25}	$0.1 + (0.9)(-0.0065)(0) = 0.1$
w_{34}	$-0.5 + (0.9)(-0.0087)(1) = -0.508$
w_{35}	$0.2 + (0.9)(-0.0065)(1) = \cancel{-}0.194$

d) Calculation for bias updating:

Bias	New Value
θ_6	$0.1 + (0.9)(0.1311) = 0.218$
θ_5	$0.2 + (0.9)(-0.0065) = 0.194$
θ_4	$-0.4 + (0.9)(-0.0087) = \cancel{0} -0.408$

37 Cross-validation :

Cross validation also referred to as out of sampling technique is a resampling technique used to evaluate machine learning models and access how the model will perform for an independent dataset. K-fold and stratified k-fold cross validations are the most used techniques. Time series cross-validation works best with time series related problems.

(38) Bootstrapping:

Bootstrapping is a statistical technique for estimating quantities about a population by averaging estimates from multiple small data samples. This is done by repeatedly taking small samples, calculating the statistic and taking the average of the calculated statistics.

A further improvement to address the pessimistic bias of the bootstrap. This pessimistic bias in the classic bootstrap method can be attributed to the fact that the bootstrap only contain approximately 63.2% of the unique samples from the original dataset. This is the significance of 0.632 bootstrapping technique. It is called 0.632 bootstrapping because 63.2% unique samples it contains.

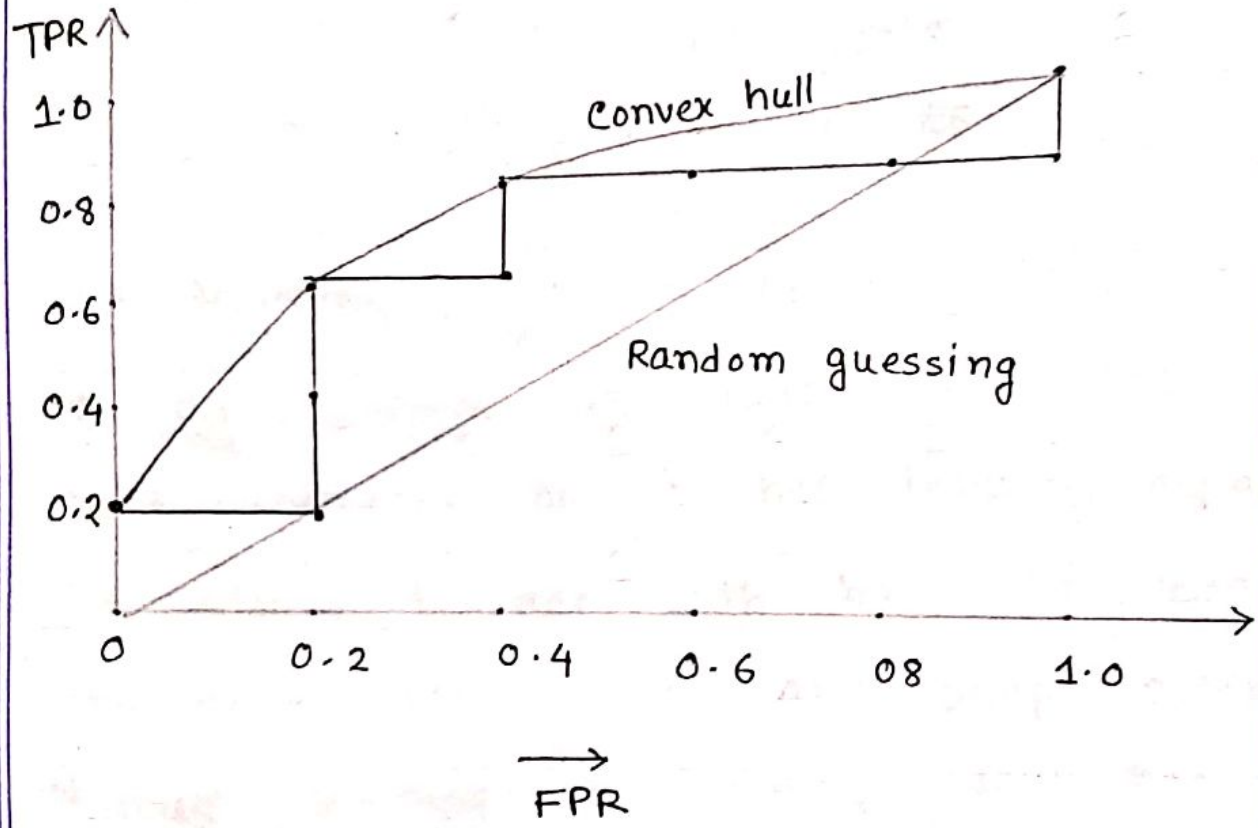
39

Tuple #	Class	Probability	TP	FP	TN	FN	TPR	FPR
1	P	0.95	1	0	5	4	0.2	0
2	N	0.85	1	1	4	4	0.2	0.2
3	P	0.78	2	1	4	3	0.4	0.2
4	P	0.66	3	1	4	2	0.6	0.2
5	N	0.60	3	2	3	2	0.6	0.4
6	P	0.55	4	2	3	1	0.8	0.4
7	N	0.53	4	3	2	1	0.8	0.6
8	N	0.52	4	4	1	1	0.8	0.8
9	N	0.51	4	5	0	1	0.8	1.0
10	P	0.40	5	5	0	0	1.0	1.0

$$TPR = TP/P = TP/TP+FN$$

$$FPR = FP/P = FP/TP+FN$$

$P = 5$, $N = 5$



ROC curve

(40) Basic types of agent program in any intelligent system are :

i) Simple reflex agent
ii) Model based reflex agent
iii) Goal based agents
iv) Utility based agents

A learning agent in AI is the type of agent which can learn from its past experience or it has learning capabilities. It starts to act with basic knowledge and then able to act and adapt automatically, through learning. A learning agent has mainly four conceptual components, which are:

i) Learning element - It is responsible for making improvements by learning from the environment.

ii) Critic: Learning element takes feedback from critic which describes how well the agent is doing with respect to a fixed performance standard.

iii) Performance Element: It is responsible for selecting external action.

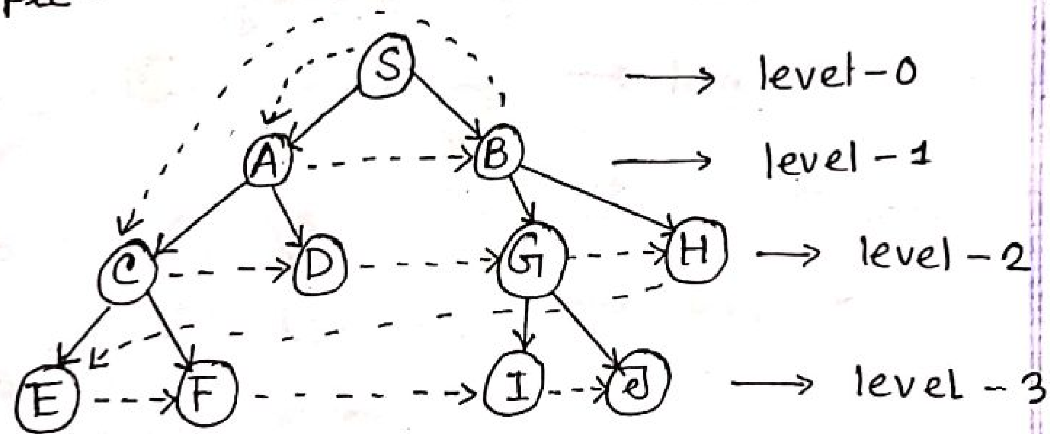
iv) Problem generator: This component is responsible for suggesting actions that will lead to new and informative experiences.

Incorporating these conceptual concepts, basic agent programs can be converted into learning agents.

41 i) Breadth First Search-

Breadth first search algorithm searches breadthwise in a tree or graph. It starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level.

Example-

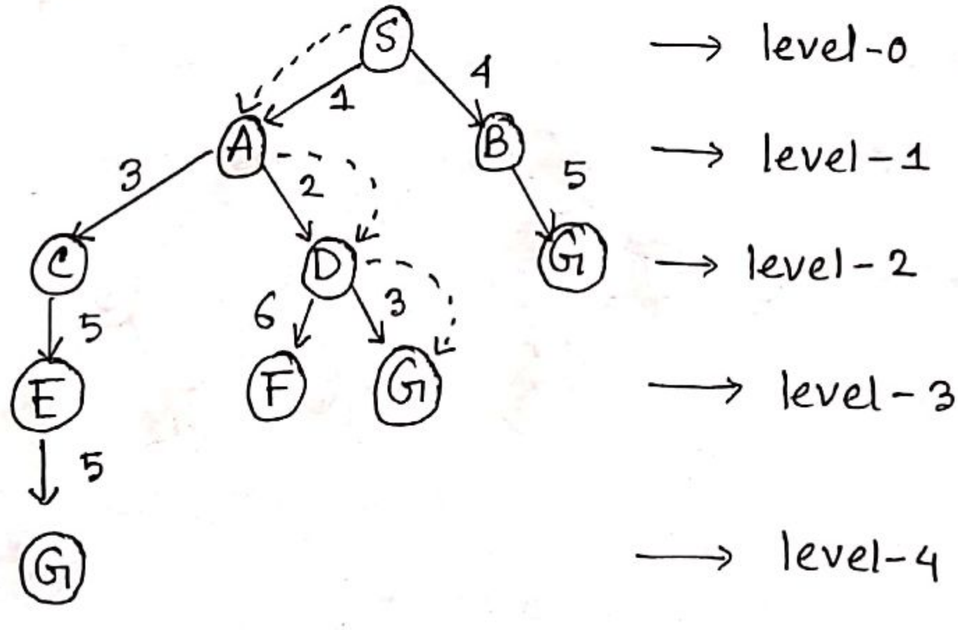


Path: S → A → B → C → D → G → H → E → F → I → J

ii) Uniform cost search-

Uniform cost search is a searching algorithm used for traversing a weighted tree or graph. The primary goal is to find a path to the goal node which has the lowest cumulative cost.

Example:

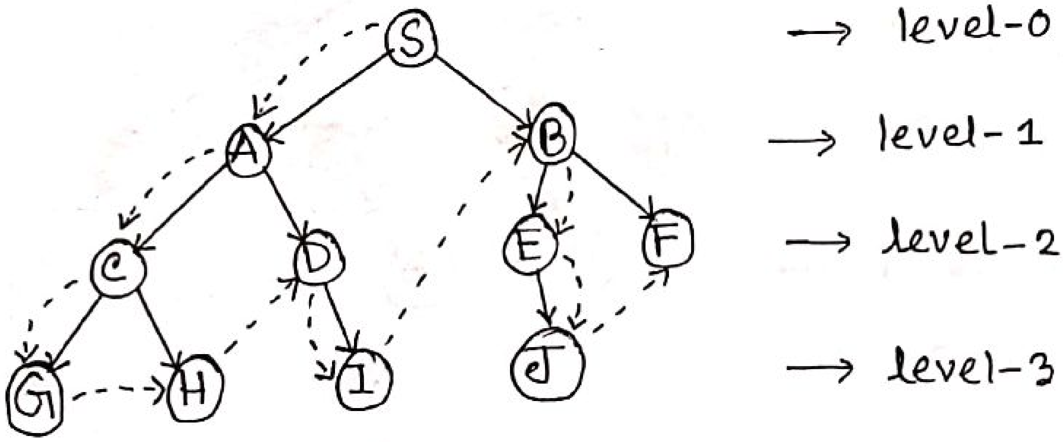


Path: S → A → D → G

iii) Depth First Search :

Depth first search is a recursive algorithm for traversing a tree or graph data structure. It starts from the node and follows each path to its greatest depth node before moving to the next path.

Example -



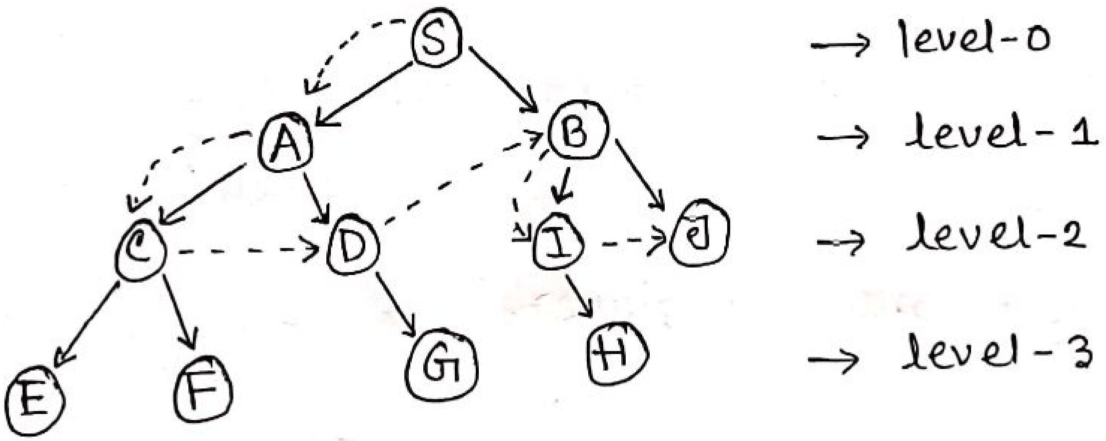
Path:

S → A → C → G → H → D → I → B → E → J → F

iv) Depth Limited Search :

A depth limited search algorithm is similar to depth first search with a predetermined limit. The node at the depth limit will treat as it has no successors nodes further.

Example -



Path : S → A → C → D → B → I → J

(42) There are different types of agents in AI. PEAS system is used to categorize similar agents together. The PEAS system delivers the performance measure with respect to the environment, actuators and sensors of the respective agent. PEAS stands for Performance measure, environment, actuator, sensor.

The PEAS ~~ea~~ description are explained below :

a) Automated taxi -

* Performance measure -

    Safe, fast, legal, comfortable trip, maximize profit

* Environments -

    Roads, other traffic, pedestrians, customers.

Actuators -

steering, accelator, brake, signal, horn, display.

Sensors -

Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboards, accelometer.

b) Medical diagnosis system -

Performance measure -

Healthy patient, minimize costs, lawsuits.

Environment -

Patient, hospital, staffs.

Actuators -

Display, questions, tests, diagnosis, treatments, referrals.

Sensors -

Keyboard entry of symptoms, findings, patient's answers.

c) Part - picking robot -

Performance measure-

Percentaage of parts in correct bins

Environment-

Conveyer belt with parts, bins.

Actuations -

Jointed arm and hand.

Sensors-

Camera, joint angle sensors.

d) Refinery controller:

Performance measure-

maximize purity, yield, safety.

Enviroment-

refinery, operator

Actuations -

Valves, pumps, heater, display.

Sensors -

Temperature, pressure, chemicals sensors

(82)

(43) The properties of task enviroments are explained below :

Fully observable vs. partially observable:
If an agent's sensors give it access to the complete state of the enviroment at each point in time, then the task environment is fully observable. A task environment is effectively fully observable if the sensors detect all aspects that are relevant to the choice of action. An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data.

Deterministic vs. stochastic:
If the next state of the environment is completely determined by the current state and the action executed by the agent, then the environment is deterministic, otherwise it is stochastic.

Episodic vs. sequential –

In an episodic task environment, the agent's experience is divided into atomic episodes. Each episode consists of the agent perceiving and then performing a single action.

In sequential environments, the current decision could affect all future decisions.

Discrete vs. continuous –

The discrete / continuous distinction can be applied to the state of the environment, to the way time is handled and to the percepts and actions of the agent. For example, a discrete-state enviroment such as a chess game has a finite numbers of distinct states. Chess also has a discrete set of percepts and actions.

Single agent vs. multiagent -

An agent solving a crossword puzzle by itself is clearly in a single agent environment, whereas an agent playing chess is in a two agent environment.

The hardest case is partially observable, stochastic, sequential, dynamic, continuous and multiagent.

(44) A well defined problem can be described by following characteristics:

- Initial state
- Operator or successor function-
  for any state x returns s(x), the set of states reachable from x with one action.
- State space-
  all states reachable from initial by any sequence of actions
- Path-
  sequence through state space.
- Path cost-
  function that assigns a cost to a path. Cost of a path is the sum of the costs of individual actions along the path.

- Goal test -

  test to determine if at goal state

A problem solving agent is a goal-based agent. It decides what to do by finding sequence of actions that lead to desirable states. The agent can adopt a goal and aim at satisfying it.

Goal formulation based on the current situation and the agent's performance measure is the first step in problem solving.

Problem formulation is the process of deciding what actions and states to consider given a goal.

45 Real world problems:

Route-finding problem:

Route-finding problem is defined in terms of specified locations and transitions along links between them. These algorithm are used in a variety of application such as routing in computer networks, military operations planning and air line travel planning systems.

Example - Airline travel problem.

Touring problem:

Touring problems are closely related to route finding problems, but with an important difference. This problem's goal test would check whether it has reached all the states once and finally reached the destination.

Example - The travelling salesperson problem.

VLSI Layout -

A VLSI layout problem requires positioning millions of components and connections on a chip to maximize manufacturing yields and minimize area, circuit delays and stray capacitance.

Automatic Assembly Sequencing -

The aim in this problem is to find the order in which to assemble the parts of some objects. Another important problem is protein design, in which the goal is to find ~~the~~ a sequence of amino acids that will be folded into a 3-D protein with the right properties to cure some disease.

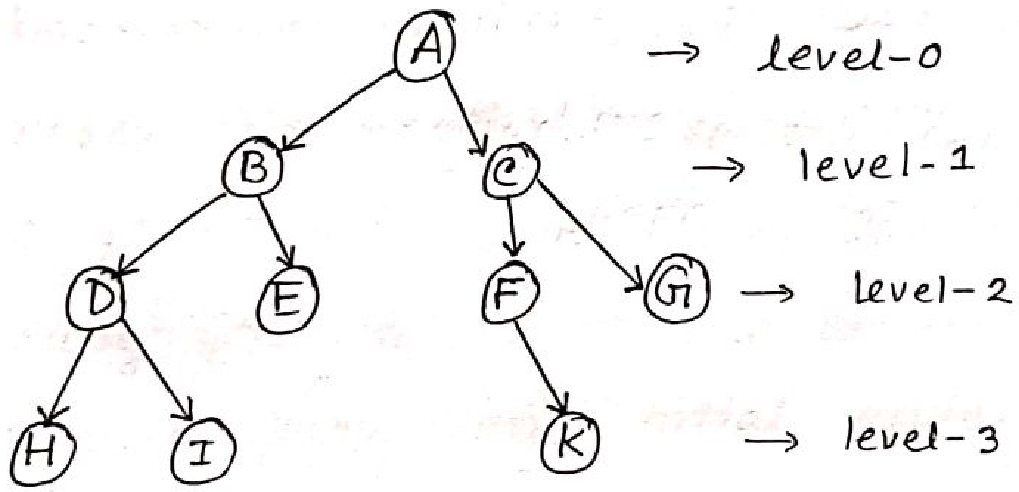
Internet Searching -

In recent years, there has been increased demand for software robots that perform internet searchings, looking for answers to questions, for related informations, or for shopping deals. The searching techniques consider internet as a graph nodes connected by links.

(47) Iterative Deepening Search:

The iterative deepening algorithm is a combination of DFS and BFS algorithms. This search algorithm finds out the best depth limit and does it by gradually increasing the limit until goal is found. This algorithm perfors depth first search up to a certain "depth limit" and it keeps increasing the depth limit after each iteration until the goal node is found.

Example :



1st iteration → A

2nd iteration → A, B, C

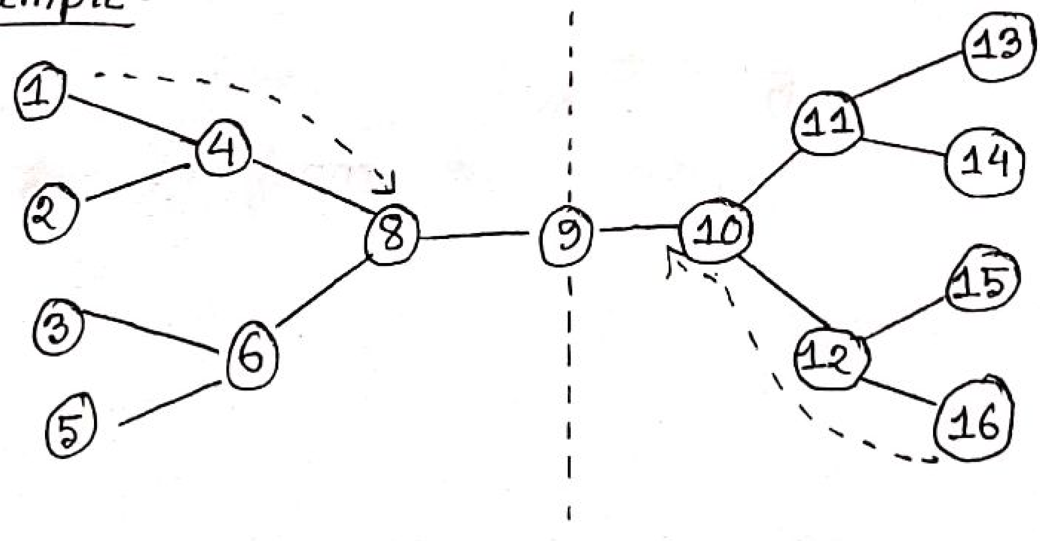
3rd iteration → A, B, D, E, C, F, G

4th iteration → A, B, D, H, I, E, C, F, K, G

(48) Bidirectional Search:

Bidirectional search algorithm runs two simultaneous searches, one from initial state called as forward-search and other from goal node called a backward search, to find the goal node. Bidirectional search replaces one single search graph with two small subgraphs in which one starts the search from an initial vertex and other starts from goal vertex. The search stops when these two graphs intersect each other.

Example -



(49) The algorithm's performance can be measured in four ways :

i) Completeness :

Is the algorithm guaranteed to find a solution when there is one?

ii) Optimality :

Does the strategy find the optimal solution?

iii) Time complexity:

How long does it take to find a solution?

iv) Space Complexity:

How much memory is needed to perform the search?

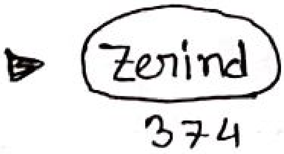
Algorithm	Completeness	Optimality	Time Complexity	Space Complexity
1) Breadth First Search	Complete	Optimal if path cost is non decreasing	$O(b)^d$	$O(b^d)$
2) Depth First Search	Complete within finite space	Non-optimal	$O(n^m)$	$O(bm)$
3) Depth Limited Search	Complete if the solution is above depth limit	Not optimal	$O(b^l)$	$O(b \times l)$
4) Iterative Deepening Search	Complete	Optimal	$O(b^d)$	$O(bd)$
5) Uniform Cost Search	Complete	Optimal	$O(b^{1+C^*/E})$	$O(b^{1+(C^*/E)})$
6) Bi directional Search	complete	Optimal	$O(b^d)$	$O(b^d)$

50

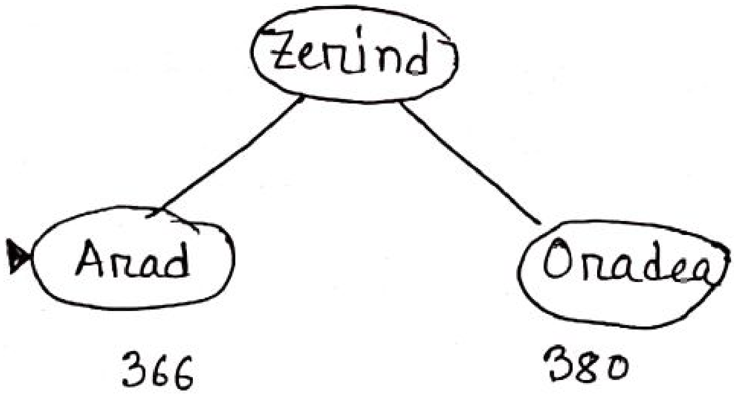
Greedy best first search tries to expand the node that is closest to the goal, on the grounds that is likely to be a solution quickly. It evaluates the nodes by using heuristic function $f(n) = h(n)$

Example - path calculation from ~~Arad~~ Zerind to Bucharest using greedy best-first search.

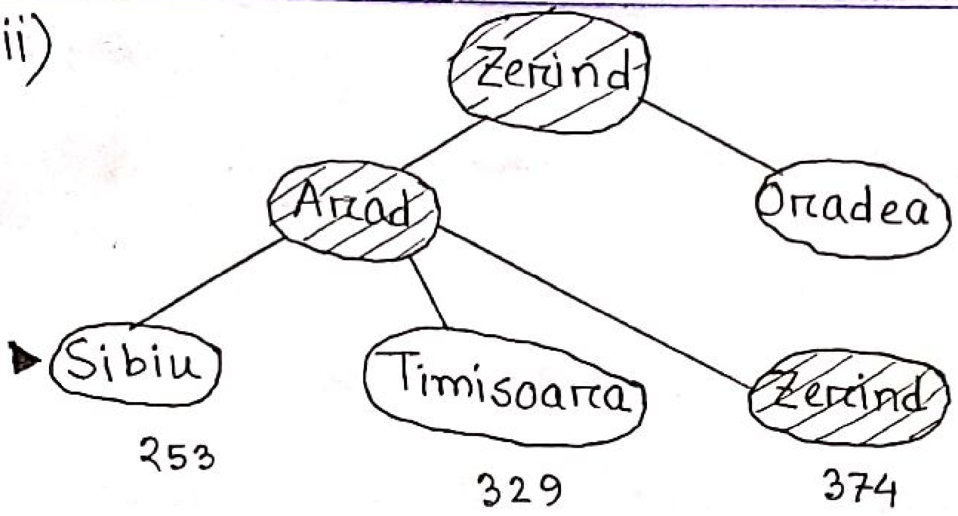
i)



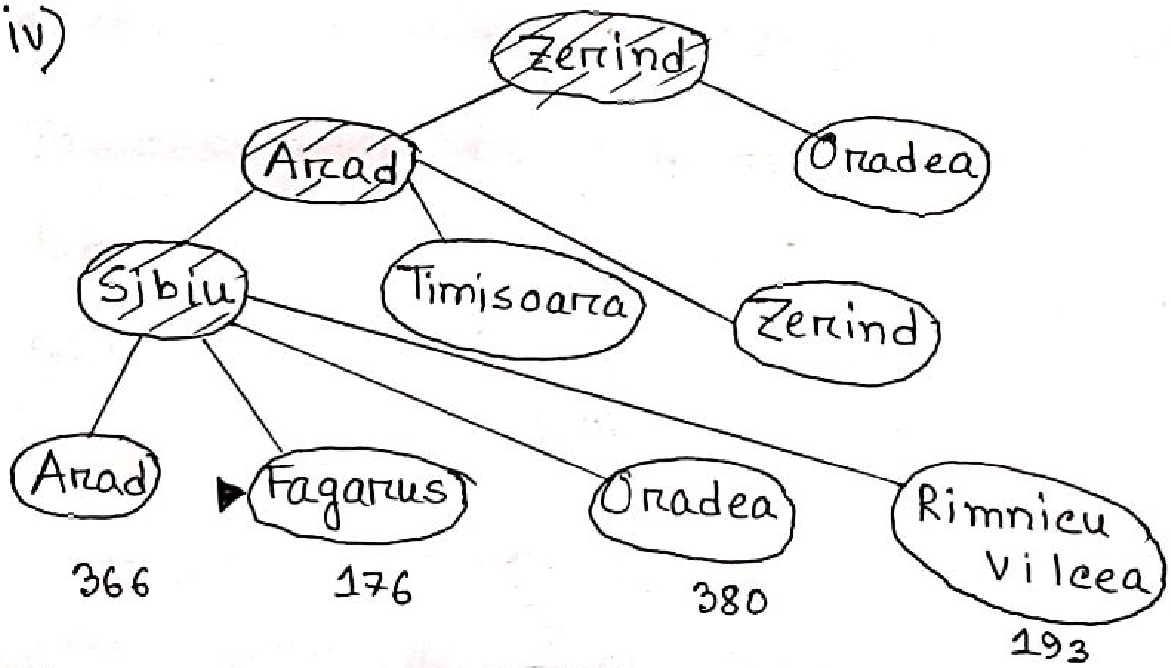
ii)



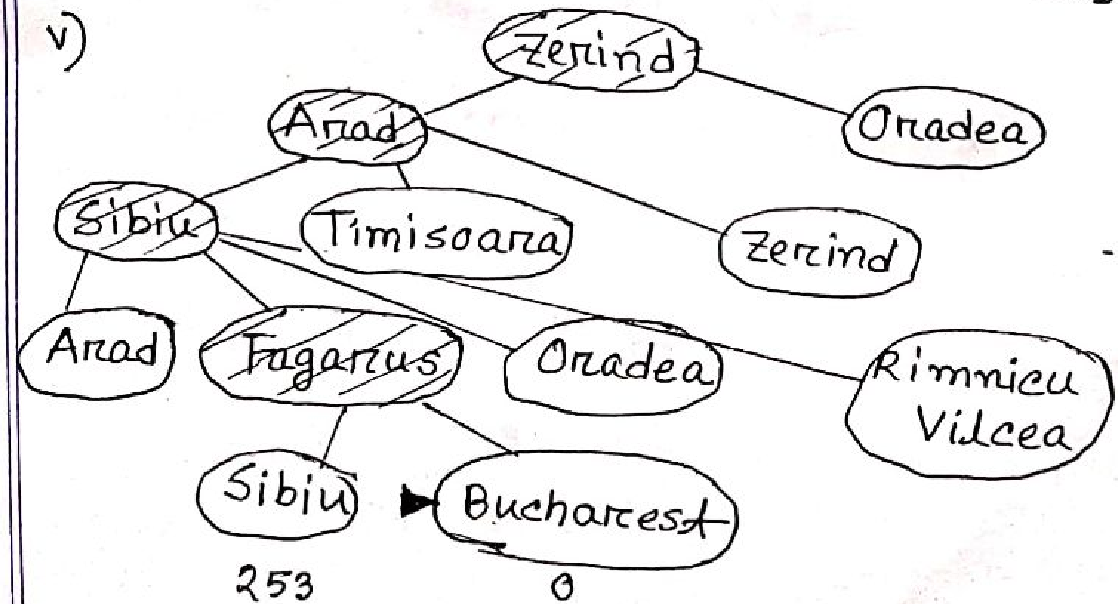
iii)


Zerind
Arad
Oradea
Sibiu
Timisoara
Zerind
253
329
374

iv)


Zerind
Arad
Oradea
Sibiu
Timisoara
Zerind
Arad
Fagarus
Oradea
Rimnicu Vilcea
366
176
380
193

v)


Zerind
Arad
Oradea
Sibiu
Timisoara
Zerind
Arad
Fagarus
Oradea
Rimnicu Vilcea
Sibiu
Bucharest
253
0

Path -

Zerind → Arad → Sibiu → Fagarus → Bucharest.

The first node to be expanded from Zerind will be Arad, because it is closer to Bucharest. Then Sibiu and Fagarus will be expanded. Fagarus in turn generates Bucharest, which is the goal.

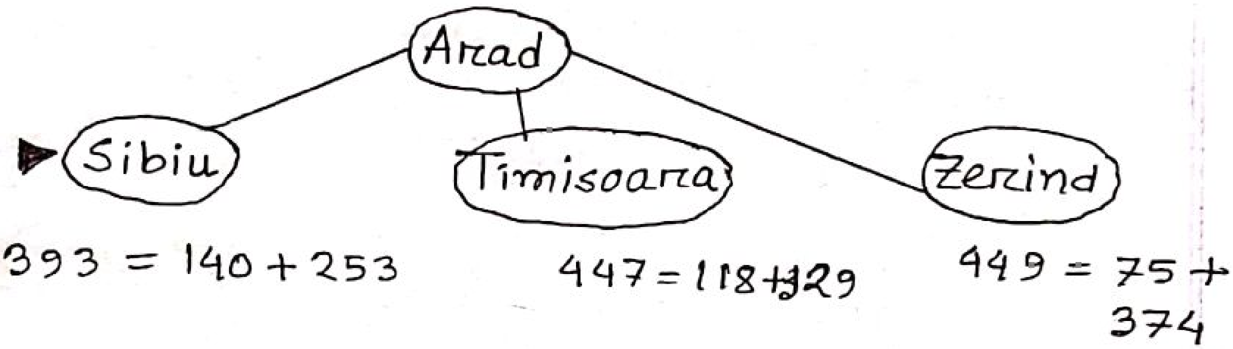
(51) A^* Search is the most widely used form of best-first search. The evaluation function $f(n)$ is obtained by combining

1) $g(n)$ = the cost to reach the node
2) $h(n)$ = the cost to get from the node to the goal : $f(n) = g(n) + h(n)$

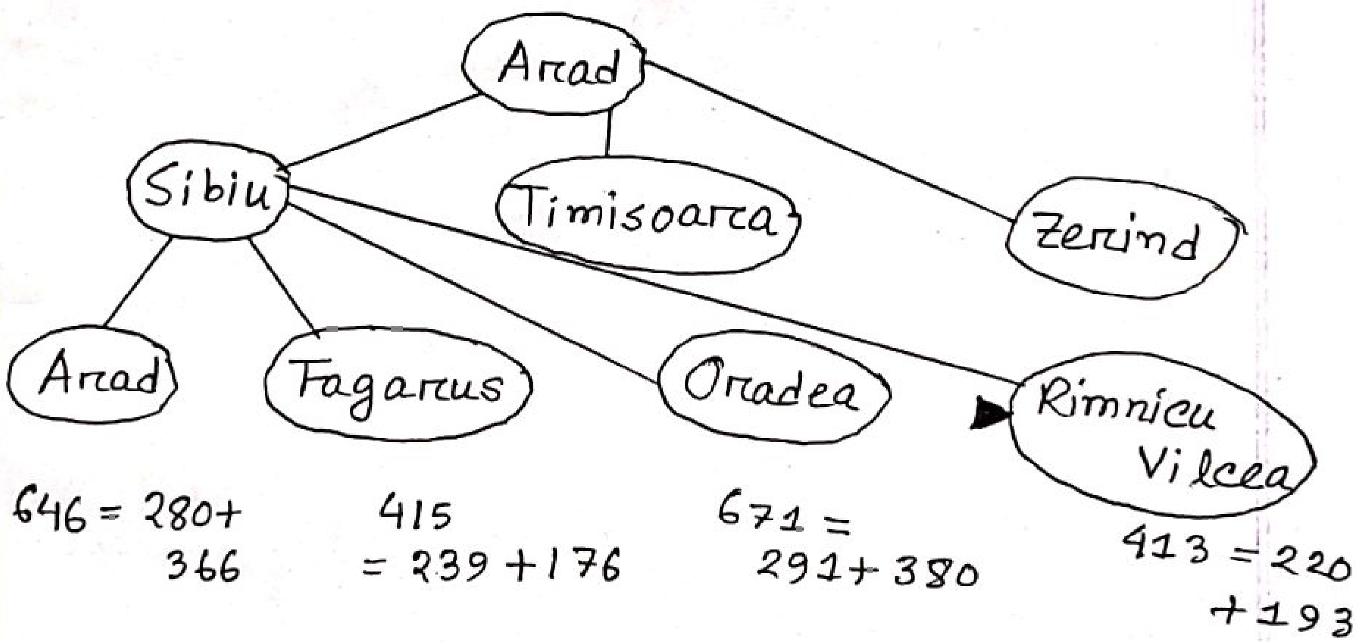
a) The initial state - ▶ Arad

366 = 0+366

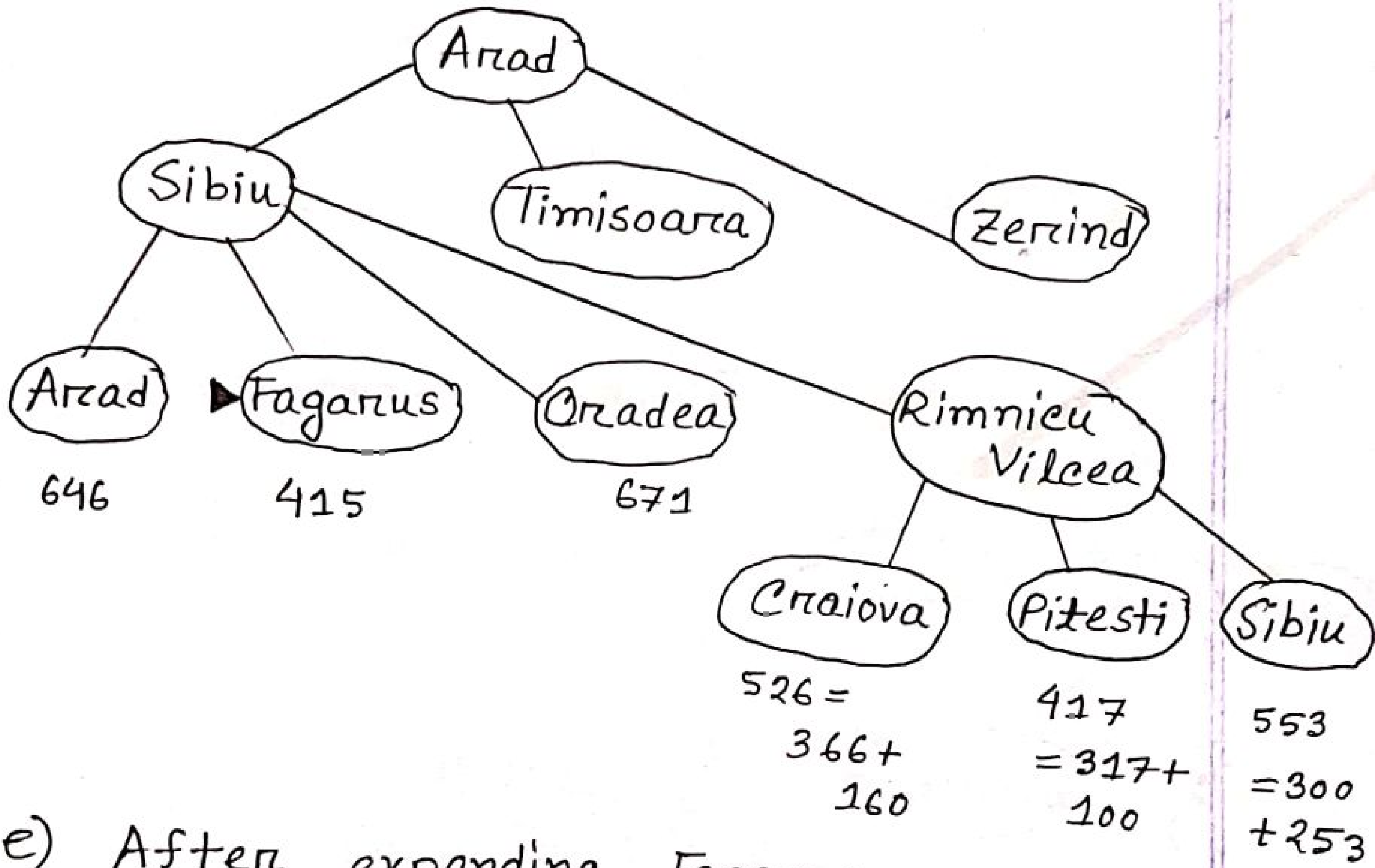
b) After expanding Arad -



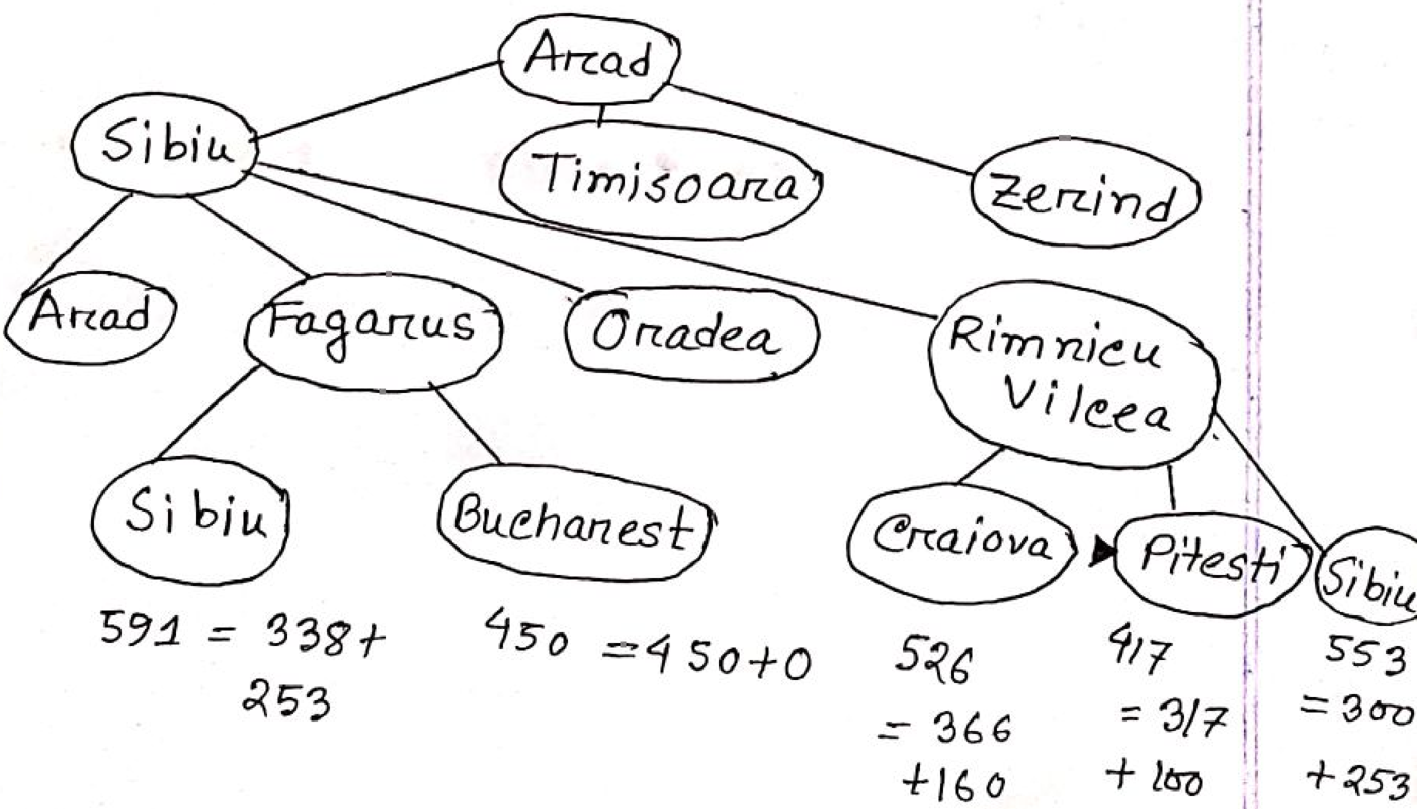
c) After expanding Sibiu -



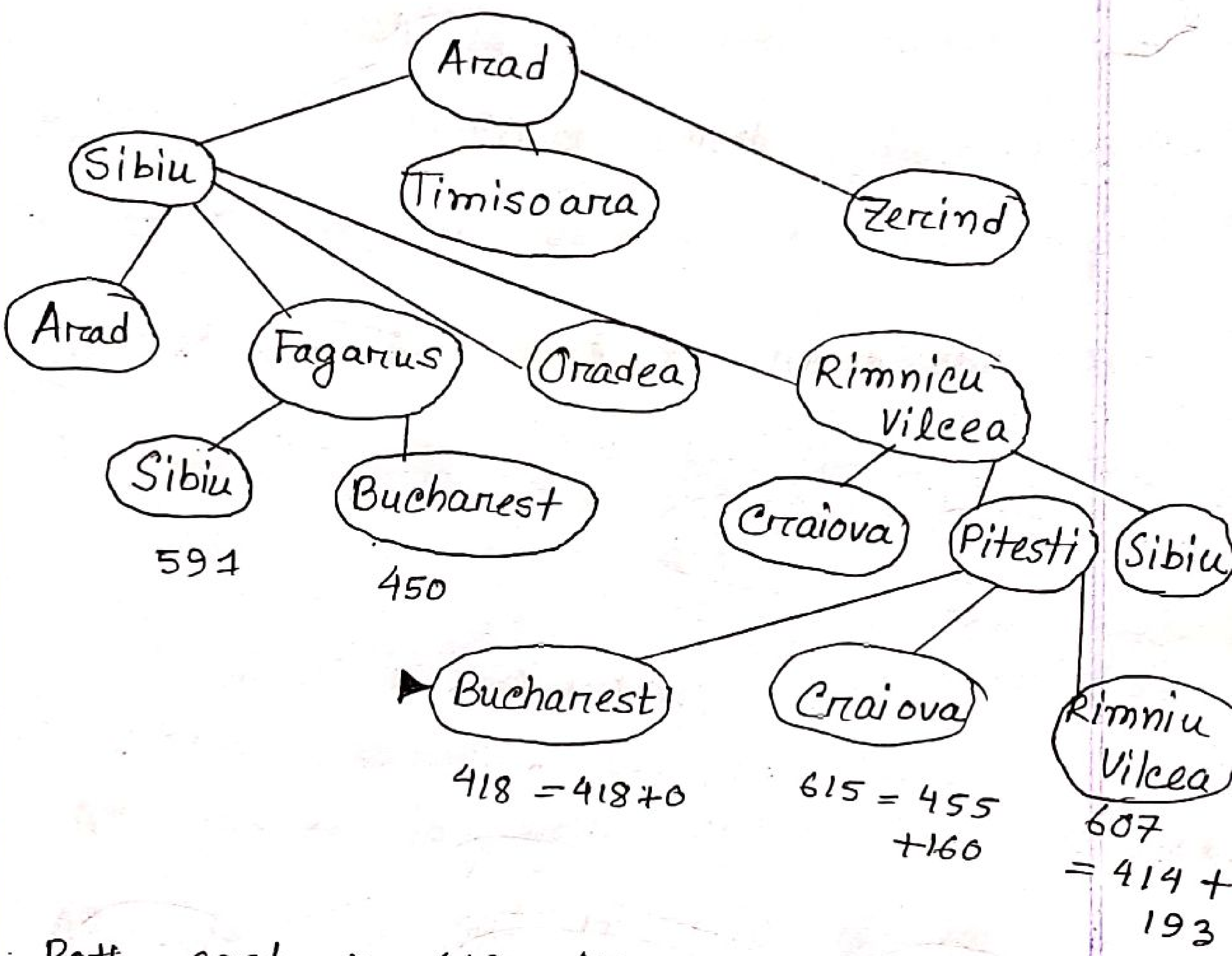
d) After expanding Rimnicu Vilcea -



e) After expanding Fagarus -



f) After Expanding Pitesti –



Path cost → 418 = 418+0.

(52) a) Recursive Best First Search (RBFS)

This algorithm mimics the operation of standard best-first search, but use only linear space. ~~Ru~~ It runs similar to recursive depth first search, but rather than continuing indefinitely down the current path, it uses the f-limit variable to keep track of the best alternative path available from any ancestor of the current node. If the current node exceeds this limit, the recursion unwinds back to the alternative path. As the recursion unwinds, RBFS replaces the f-value of each node along the path with the best f-value of its children. In this way, it can decide whether it is worth reexpanding a forgotten subtree.

b)

Heuristic functions :

A heuristic function or simply a heuristic is a function that ranks alternatives in various search algorithms at each branching step based on an available information in order to make a decision which branch is to be followed during a search.

53 a) Hill climbing -

Simple hill climbing search algorithm is a local search algorithm where greedy approach is used without any backtracking.

Steps :

1) Evaluate the inital state

2) Loop until a solution is found or there is no operator left.

3) Select and apply a new operator.

4) Evaluate the new state

5) If goal, then quit

6) If better than current state, then assign it a new current state.

It only evaluates the neighbour node at a time and selects the first one which optimizes current cost and set it as a current state.

b) Genetic Algorithm-

This algorithm simulates the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. They simuelate " survival of the fittest " among individual consecutive generation for solving a problem.

Steps -

1) Initialize population
2) Selection according to the fitness.
3) Crossover between selected
4) Perform mutation
5) Repeat cycle, till the condition of stop is true.

e) Simulated Annealing :

Simulated annealing is an algorithm which yields both efficiency and completeness. It is a variation of hill climbing in which, at the beginning of the process, some downhill move, may be made, lowering the chances of getting caught at a local maximum. Annealing is a process where the algorithm picks a random value or move instead of picking the best move. If the random value improves the state, then it follows the same path. Otherwise the algorithm follow the path which has probability of less than 1 or it moves downhill and choose another path.

d) Local Beam Search -

Unlike hill climbing, local beam search keeps track of K states rather than just one. However it suffers from lack of diversity and choose k successor proportional to state quality.

Steps -

i) It starts with K randomly generated states

ii) At each step, all the successors of all states are generated.

iii) If any one is a goal, the algorithm halts, otherwise it selects the k best successors from the complete list and repeats.

(54) Constant satisfaction problems mathematical problems defined as a set of objects whose state must satisfy a number of constraint or limitations. CPS represent the entities in a problem as a homogenous collection of finite constraints over variables, which is solved by constrain satisfaction method.

To formulate a CSP, the variables are needed to be defined in the regions X.

A constrain ~~sta~~ satisfaction problem $\{X, P, C\}$

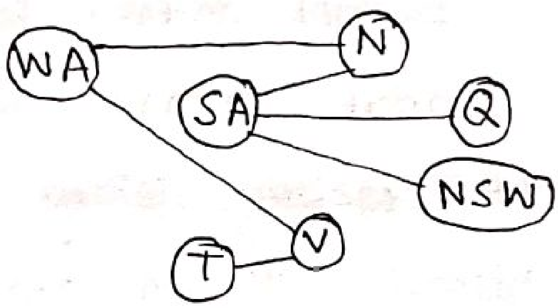
$X = \{X_1, X_2, \dots, X_n\}$ is a set of variables.

$D = \{D_1, D_2, \dots, D_n\}$ is a set of their respect domain value.

$C = \{C_1, \dots, C_m\}$ is a set of constraint.

(55) a) Constraint Graph -

The nodes of the graph correspond to variables of the problem and the arcs correspond to constraints.



b) Cryptarithmetic problem-

It is a type of constraint satisfaction problem where the game is about digits and its unique replacement either with alphabets or other symbols.

Example-

$$\begin{array}{r} \text{SEND} \\ +\ \text{MORE} \\ \hline \text{MONEY} \end{array}$$

In this example, both terms ~~so~~ SEND and MORE to bring MONEY as a result.

The representation of the assignment of the digit to the alphabets.

S	E	N	D	M	O	R	Y
9	5	6	7	1	0	8	2

c) Adversarial Search Problem-

This is a search where we examine the problem which arises when we try to plan ahead of the world and other agent are planning against us.

d) Games-

A branch of economics, views any multiagent environment as a game provided that the impact of each agent on the other is significant, regardless of whether the agents are cooperative or competitive.

56

i) The minimax algorithm-

This algorithm computes the minimax decision from the current state. It uses a simple recursive computation of the minimax values of each successor state, directly implementing the defining equations. The recursion proceeds all the way down to the leaves of the tree and then the minimax values are backed up through the tree as the recursion unwinds. The minimax algorithm performs a complete depth first exploration of the game tree. If the maximum depth of the tree is m and there are b legal moves at each point, then the time complexity is $O(b^m)$ and space complexity is $O(bm)$

ii) Alpha - Beta Pruning -

The problem with minimax search is that the number of game states it has to examine is exponential in the number of moves. But the exponent can not be eliminated but it can be effectively cut in half. The technique is called alpha beta pruning, when applied to a minimax tree, it returns the same move as minimax would, but prunes away branches that can not possibly influence the final decision.

- α : the value of the best choice at any choice point along the path of MAX.

- β : the value of best choice at any choice point along the path of MIN.

Alpha Beta search updates the values of α and β as it goes along and prunes the remaining branches at a node as soon as the value of current node is known to be worse than the current α and β value for MAX and MIN, respectively.