AI

Assignment-2

18124011600l
Vishwas Acharya

Page No. :
Date : / /

1) What is state space of a problem?

- Define STATE : A state is a representation of problem elements at a given moment.

- Define STATE SPACE : A state space is the ~~state~~ set of all states reachable from the initial state.

- A state space forms a graphs in which the nodes are states and the arcs between nodes are actions.

- In state space, a path is a sequence of states connected by a sequence of actions.

- The solutions of a problem is part of the graph formed by the state space.

- Example of state space search

(i) Problem of playing chess
(ii) Water Jug Problem
(iii) 8-puzzle problem.

2) Describe Breadth First Search commad on the optimality of this method.

Ans - This is an exhaustive search technique
- The search generate all nodes at a

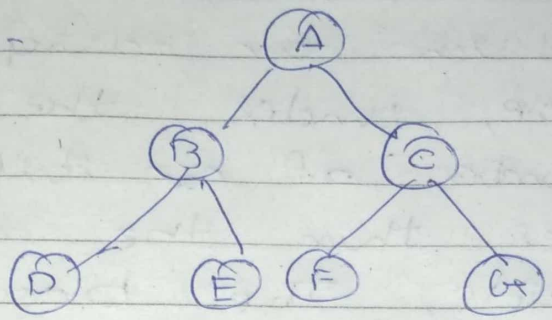
particular level before proceeding to the next level of the tree.
- It is a graph traversal algorithm that starts traversing the graph from root node & explores all the neighbouring nodes.
- Then, it selects the nearest node & explore all the unexplored nodes.
- The algorithm follows the same process for each of the nearest node until it finds the goal.
- It can be implemented using FIFO queue data structure
- This method provides shortest path to the solution
- The total no. of nodes created in worst case is $b + b^2 + b^3 + \dots + b^d$ where d = depth.

- **Algorithm:-**

(i) Create a variable called NODE-LIST & set it to initial state
(ii) Until a good state is found or NODE-LIST is empty do:
↳ Remove the first element from NODE-LIST & call it E. If NODE-LIST was empty, quite
↳ For each way that each rule can match the state described in E do:

→ Apply the rule to generate a new state
→ If the new state is a good state, quit & return this state
→ otherwise, add the new state to the end of NODE-LIST

- Example :-



- Time complexity :- Time complexity of BFS algorithm can be obtained by the new of nodes traversed in BFS until the shallowest Node, where the d(depth) of shallowest solution & b is a node at every state

$$T(b) = 1 + b^2 + b^3 + \dots + b^d = O(b^d)$$

- Space complexity :- of BFS algorithm is given by the memory size of frontier which is $O(b^d)$

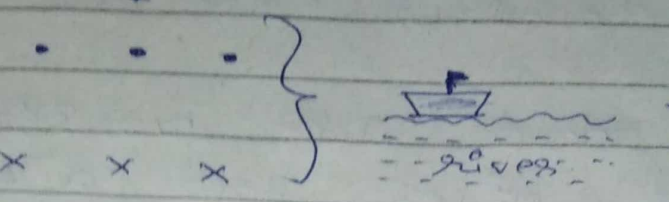
- Completeness :- BFS is complete, which means if the shallows goal node is at some finite depth then BFS will find a solution.

- Optimality: BFS is optimal if path cost is a non-decreasing function of the depth of the node.

3b In the Missionaries and cannibals Problem, three missionaries and three cannibals must cross a rives using a boat which can carry at most two people, under the constraint that, Number of cannibals should be lesser than the missionaries on either side, The boat cannot cross the rives by itself with no people on board. For the above mentioned problem, describe state space representation, action. start and end state.

Ans - In this problem, 3M & 3C must cross a rives using a boat which can carry at most two people, under the constraint that, for both banks, that the missionaries present on the bank cannot be outnumbered by ~~com~~ cannibals.

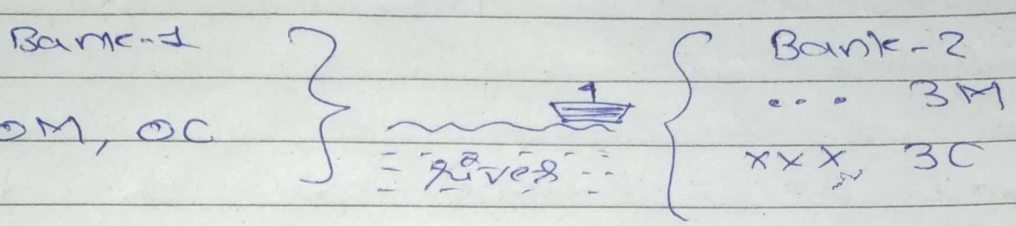
- The boat cannot cross the rives by itself with no people on board.



- First let m cosides that both the missionaries (M) & cannibals (C) are on the same side of the river. left - Right.
Initially the positions are: 0M, 0C & 3M, 3(C)'

- Now, let's send 2C to left ~~to~~ of Bank:
0M, 2(C)(B) & 3M, 1C
- Send one cannibals from left to right : 0M1, 1C & 3M, 2(C) (B)
- Now send the 2 remaining cannibals to left 0M, 3(C) (B) & 3M, 0C
- Send 1C to the left : 0M, 2C & 3M, 1 (C B)
- Now send 2 missionaries to the left : 2M, 2C(B) & 1M1, 1C
- Send 1M & 1C to the left : 1M, 1C & 2M, 2C(B)
- Send 2M to left : 3M1, 1C(B) & 0M, 2C
- Send 1C to right : 3M1, 0C & 0M, 3C(B)
- Send 2C to left : 3M, 2C(B) & 0M, 1C
- Send 1C to right : 3M, 1C & 0M, 2C(B)

- Send 2C to left: 3M, 3C(B) & 0M, 0C
- Here, (B) shows the positions of the boat after the actions is performed.
- Therefore all the missionaries & cannibals have crossed the rives safety.



4) Describe heuristic function for the travelling salesman problem.

Ans - The traveling salesman problem abide by a salesman & set of cities.
- The salesman head to visit every one of the cities starting from a certain one (e.g. the hometown) & to refrain to the somecity.
- The challenge of the problem is that the traveling salesman needs to minimize the total length of the trip.
- Suppose the cities are $x_1, x_2 \ldots x_n$ where cost C_{ij} denotes the cost of travelling from city x_i to x_j.

- The travelling saleiman problem is to find a route starting & ending at x_y that will take in all cities with the minimum cost.

- Heuristic function maps from problem state descriptions to measures of desirability, usually represented as numbers.
- well designed heuristic functions can play an important part in efficiently guiding a search process towards a solution.
- The objective of the TSP is to find the lowest cost route that satisfies the problem's four main constraints, specified below:

Minimize $\sum_{(i,j) \in A} c_{ij} \cdot x_{ij}$

Subject to : $\sum_{i \neq \{j\}} x_{ij} = 1, j \in X \quad - (1)$

$\sum_{j \in V \setminus \{i\}} x_{ij} = 1, \; i \in V \quad - (2)$

$\sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1; \; S \subseteq V; \; |S| \geq 2 \quad - (3)$

$x_{ij} \in \{0, 1\}, \; (i,j) \in A \quad - (4)$

- TSP can be divided into two general types
  (i) Symmetric TSP [STSP]
  (ii) ASymmetric TSP [ATSP]

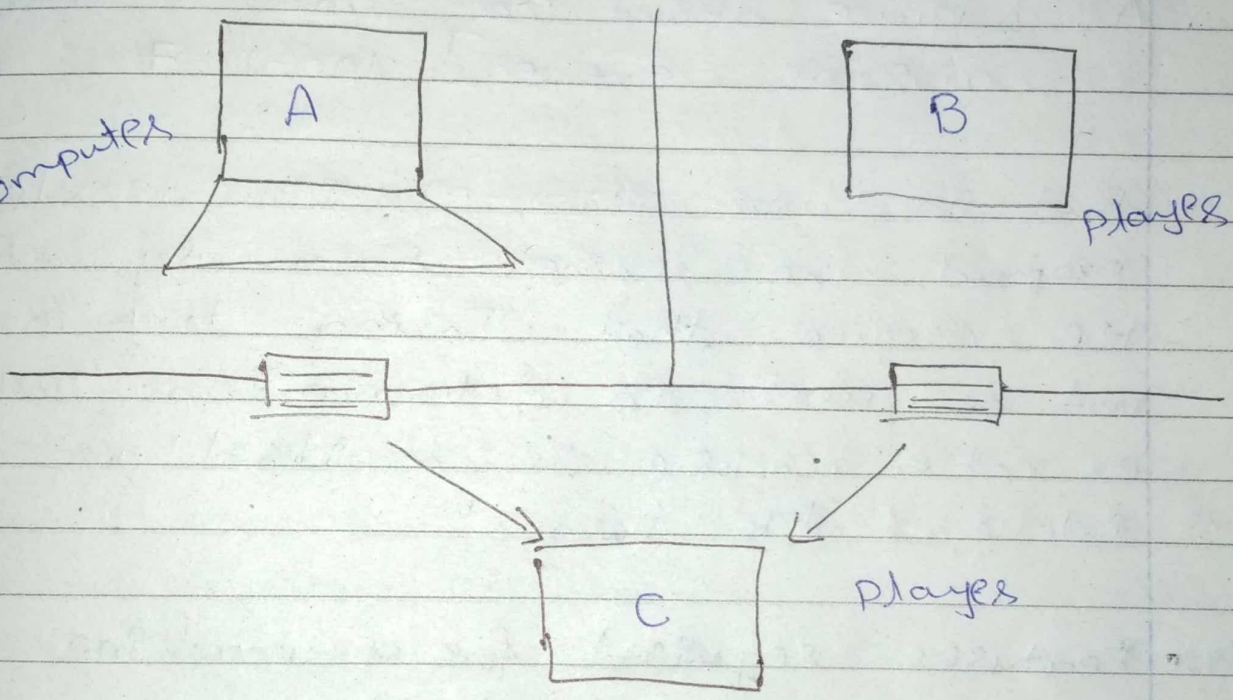
- ATSP is usually related to intracity problems.
- By contrast, the STSP is mostly for intercity problems, usually with roughly symmetric roads
- One way to create an effective heuristic is to remove one or more of the underlying problem's constraints & then modify the solution to make it conform to the constrain after the fact or the otherwise use it to inform your heuristic.

5) Discuss Turing Test

Ans - The Turing test developed by Alan Turing (computer scientist) in 1950.
- He proposed that Turing test is used to determine whether or not computer (machine) can think intelligently like human?
- Imagine a game of 3 players having two human & one computer, an interrogation (as human) is isolated from other two players
- The interrogates job is to try &

figure out which one is human & which one is computer by asking questions from both of them.

- To make the things harder computer is trying to make the interrogators guess wrongly.
- In other words computer would try to indistinguishable from human as much as possible.



- The standard interpretation of the Turing test, in which player C, the interrogator, is given the task of trying to determine which player A or B - is a computer & which is a human. The interrogater is limited to using the responses to written questions to make the determination

- The conversation between interrogeter & computes would be like this!

C [Interrogeution]: Are you a computes?
A [computes]: No.
C: Multiply one large numbes to anothes 158745887 * 56757647
A: After a long pause, an incorrect answes
C: Add 5478012, 4583145
A: [Pause about 20 second & than give as answes] 10061157

- But in yeas 1980, Mr John searle proposed the "chinese room argument".
- He argued that Turing test could not be used to determination "whethe os not a machine is considered as intelligent like humans".

- Features required for a machine to pass the Turing test:

- Natural language processing: NLP is required to communicate with Interrogator in genesal human language like english.

- Knowledge representation: To stose & retrive information during the test

- Automated reasoning : To use the previously stored information for answering the questions
- Machine learning: To adopt new changes & can detect generalized patterns
- Vision (For total Turing test): To recognize the interrogates actions & other objects during a test
- Motor control (For total Turing test): To act upon ~~of~~ objects if requested.

6b What is the significance of the 'Turing Test' in AI? Explain how it is performed.

Ans The turing test is a deceptively simple method of determining whether a machine can demonstrate human intelligence, if machine can engage in a conversation with a human without being detected as a machine, it has demonstrated human intelligance.

- It has become a fundamental motivator in the theory & development of AI.
- In 1950, Alon Turing introduced a test ~~test~~ to check whether a machine can think like a human or not, this test is known as the Turing test

- In this test, Turing proposed that the computer can be said to be an intelligent if it can mimic human response under specific conditions.

- Rapid advances in computing are now visible in many aspects of our lives
- we have program that translate one language to another in the blink of an eye; robots that clean an entire room in retirement portfolios & wearable ~~desis~~ devices that track our health & fitness levels
- All of these have become relatively mandane
- At the forfront of disruptive technology now are the pioneers in the development of artifical intelligent.

7) Discuss & Analyze Towes of Hanoi problem with respect to the seven problem characteristics.

Ans - This is a fun puzzle game where the objective is to move an entire stack of disks from the source position to mather position.
- There simple rules are followed.
(i) Only one disk can be moved at a time.
(ii) Each move consists of taking the

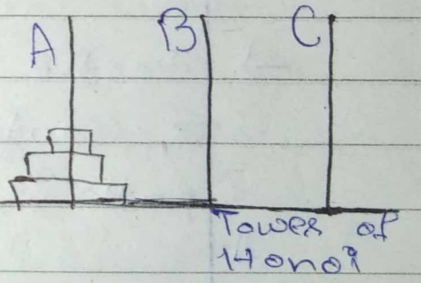
upper disk from one of the stacks & placing it on top of another stacks. In other words, a disk can only be moved if it is the uppermost disk on a stack.

(iii) No larger disk may be placed on top of a smaller disk.

- Now let's try to imagine a scenario. Suppose we have a stack of three disks. Our job is to move this stack from source A to destination C. How do we do this?

- Before we can get there, let's imagine there is an intermediate point B

- We can use B as a helper to finish this job. We are now ready to move on. Let's go through each of the steps.



- Move the first disk from A to C
- Move the first disk from A to B
- Move the first disk from C to B
- Move the first disk from A to C
- Move the first disk from B to A
- Move the first disk from B to C
- Move the first disk from A to C

Problem characteristics	Satisfied	Reason
→ Is the problem decomposable?	No	One game have single solution.
→ Can Solution steps be ignored or undone?	Yes	We can undo the previous move.
→ Is the problem universe predictable?	Yes	We can predict about the solution
→ Is a good solution absolute or relative?	Absolute	Absolute:- once you get one soln you do need to bother about other possible solution Relative:- once you get one solution you have to find another possible solution to chect which solution is best.
→ Is the solution a state or a path?	Path	A path to a state you have perfect rules for problem, no need to worry about logic of work
→ What is the role of knowledge	Need knowledge	lot of knowledge help to constrain the search for a solution
→ Does the tasks require interaction with a person/ human?	No.	Conservational in which there is intermediate communication between a person & the computer

8) Discuss, simulated annealing search method. Compare it with hill climbing method.

Ans - It is based on metallurgical practice by which a material is heated to a high temperature & cooled.

- A hill-climbing algorithm which never makes a move towards a lower value guaranteed to be in complete because it can get stuck on a local maximum
- And if algorithm applies a random walk by moving a successor, then it may complete but not efficient.
- Simulated Annealing is an algorithm which yields both efficiency & completness
- In mechanical term Annealing is a process of hardening a metal or glass to a high temperature then cooling gradually, so this allows the metal to reach a low-energy crystalline state.
- This can be done in two ways:
  i) Using prior knowledge about the problem to input a good starting point.
  ii) Generating a random solution
- Hill climbing always get stuck in a local maxima because downward moves aren't allowed.

- Simulated annealing is technique that allows downward steps in order to escape from a local maxima. The idea behind annealing is that, at high temperature the algorithm should jump out of a local maxima.

9) Explain state space representation using watering problem.

Ans
- We have two jugs 4 gallon & 3 gallon.
- Here the initial state is (0,0). The goal state is (2,n) for any value of n.
- State space representation: We will represent a state of the problem as a tuple (x,y) where x represents the amount of water in the 4 gallon jug & y represents the amount of water in the 3 gallon jug:

$$0 \le x \le 4 \quad \& \quad 0 \le y \le 3$$

- To solve this we have to make some assumption not mentioned in the problem.
- They are,

(i) We can fill a jug from the pump.
(ii) We can pour water out of a jug to the ground
(iii) There is no measuring devices available

	Current State	Next State
1.	(x,y) if $x<4$	$(4,y)$. Fill the 4 gallon jug
2.	(x,y) if $y<3$	$(x,3)$ Fill the 3 gallon jug
3.	(x,y) if $x>0$	$(x-d,y)$. Pours some water out of the 4-gallon jug.
4.	(x,y) if $y>0$	$(x,y-d)$. Pours some water out of the 3-gallon jug.
5.	(x,y) if $x>0$	$(0,y)$. Empty the 4-gallon jug
6.	(x,y) if $y>0$	$(x,0)$. Empty the 3-gallon jug
7.	(x,y) if $x+y>=4$ & $y>0$	$(4, y-(4-x))$. Pours water from 3 gallon jug into 4 gallon jug until the 4-gallon jug is full.
8.	(x,y) if $x+y\geq 3$ & $x>0$	$(x-(3-y), 3)$. Pours water from the 4 gallon jug into the 3 gallon jug until the 3 gallon jug is full.
9.	(x,y) if $x+y<=4$ & $y>0$	$(x+y,0)$, Pours all the water from the 3 gallon jug into the 4 gallon jug
10.	(x,y) if $x+y<=3$ & $x>0$	$(0,x+y)$, Pours all the water from the 4-gallon jug into the 3 gallon jug.
11.	$(0,2)$	$(2,0)$ Pour the 2 gallons from 3 gallon jug into the 4 gallon jug.
12.	$(2,y)$	$(0,y)$. Empty the 2 gallons in the 4 gallon jug on the ground.

- These are several sequences of operators that will solve the problem.
- One of the possible solutions is given as.

4-gallon jub	3-gallon jub	Rule Applied
0	0	2
0	3	9
3	0	2
3	3	7
4	2	5 or 12
0	2	9 or 11
2	0	-

10b Discuss A* algorithm. Also give one example to explain it.

Ans - It is best-known form of BFS. It avoids expanding paths that are already expensive but expands most promising paths first.
- A* search algorithm is one of the best & population technique used in path-finding & graph traversals.
- The algorithm uses following functions :-

$f' = g + h'$

where, f' = Heuristic function that estimates the merits of each node we generate

g = It is a measure of the cost of getting from initial state to the current node.

h' = is an estimate of the addition cost

of getting from the current node to a goal state.

- The algorithm also uses the lists : OPEN and CLOSED.
- f' represents an estimates of the cost of getting the initial state to a goal state along with the path that generated the current note.

- Worst case : $O(|E|) = O(b^d)$
- Space complexity: $O(|V|) = O(b^d)$

- Algorithm

step-1] Define a list OPEN initially OPEN consists safely of a single node the start node.

step-2] If the list is empty, return failure & exit.

step-3] Remove node n with the smallest value of f(n) from OPEN & move it to list CLOSED

step-4] Expand node n

step-5] If any successor to n is the goal node, return success & the solution by tracking the path from goal node to S.
Otherwise, go to step-6

step6] For each successor node,
↳ Apply the evaluation function f to the node.
↳ If the node has not been in either list, add it to OPEN

step-7] Go back to step-2

- Example:- Given an initial state of a & puzzle problem & final state to be reached

2	8	3
1	6	4
7		5

Initial state

1	2	3
8		4
7	6	5

Final state.

11) Explain what is meant by "production system" with respect to AI. Discuss the component of a production system.

Ans - A production system (production rule system) is a kind of cognitive architecture that is used to implement search algorithm.
- The problem solving knowledge is encoded in ~~the~~ system in the form ~~to~~ of little quanta popularly known as productions.
- It consists of two components rule & action.
- Rules recognize the condition & the actions part has the knowledge of how to deal with the condition.
- ~~The~~ In similar words, the production system in AI contains a set of rules which are defined by the left side & right side of the system.

- The left side contains a set of things to watch for (condition) & the right side contains the thing to do (action).

- Components of Production System:
- Global Database: It is the constract data structure used by the production system in AI.
  - ↳ The primary database which constrains the information necessary to successfully complete a task.
  - ↳ It has two parts.
  - (i) Temporary part contains information relevant to the current situation only whereas the
  - (ii) Permanent part contains information about the fixed actions.

- Set of Production Rules :- It operate on the global database. Each rule usually has a precondition that is either satisfied or not by the global database.
  - ↳ For ex. if a condition is met by the global database, then the production rule is applied succefully.

- Control System :- It acts as the decision-makes decides which production rule should be applied.
  - ↳ The control system stops computation

processing when a termination condition is met on the database.

12) Explain how a problem can be analyzed based on its characteristics. Analyze the game of "8-puzzle" based on these characteristics.

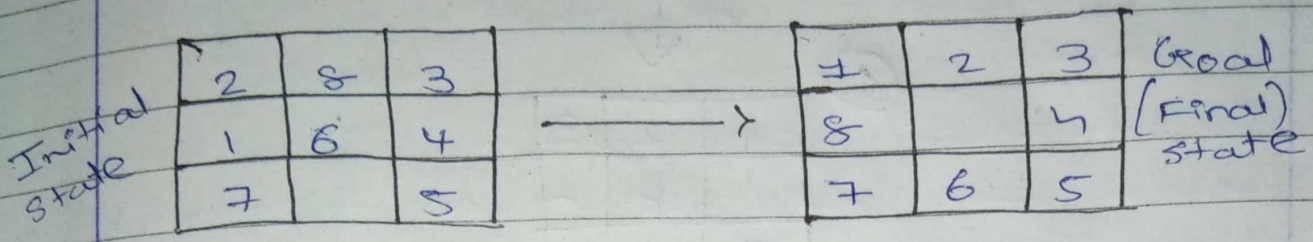
Ans - In order to choose the most appropriate problems solving method, it is necessary to analyze the problem along various key dimensions.

- Is the problem decomposable into a set of independent smaller or easier ~~suits~~ sub problem.
- Can solution steps be ignored or at least under if they prove unwise?
- Is the problem's universe predictable?
- Is a good solution to the ~~pl~~ problem obvious without comparision to all other possible solution?
- Is the desired solution a state of the world or a path to a state?
- What is the role of knowledge?
- Does the tant require interaction with a person?

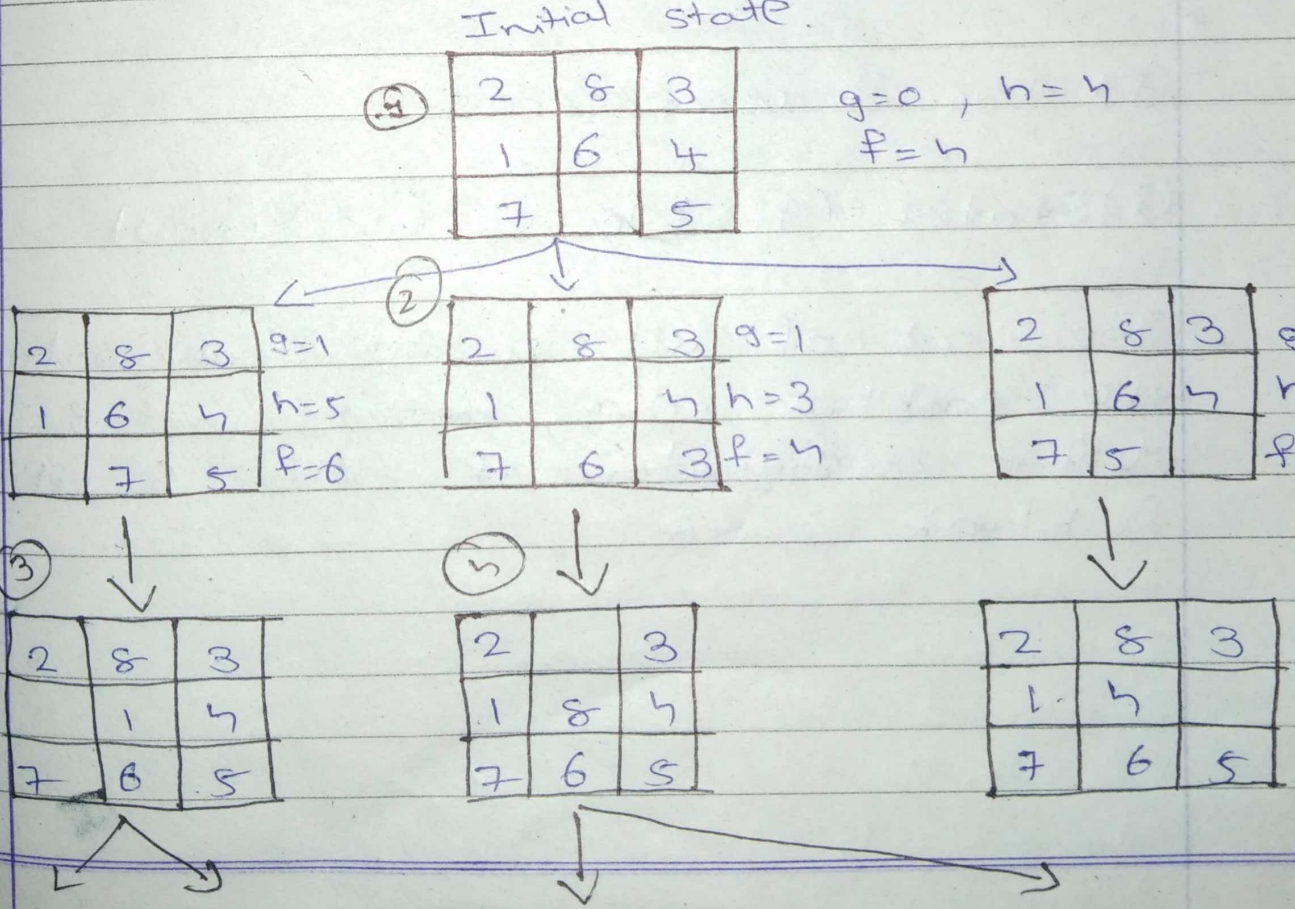
- 8-puzzle problem:- It consists of 8 numbered moveable tile set in 3x3 frame.
- One cell of the fram is always

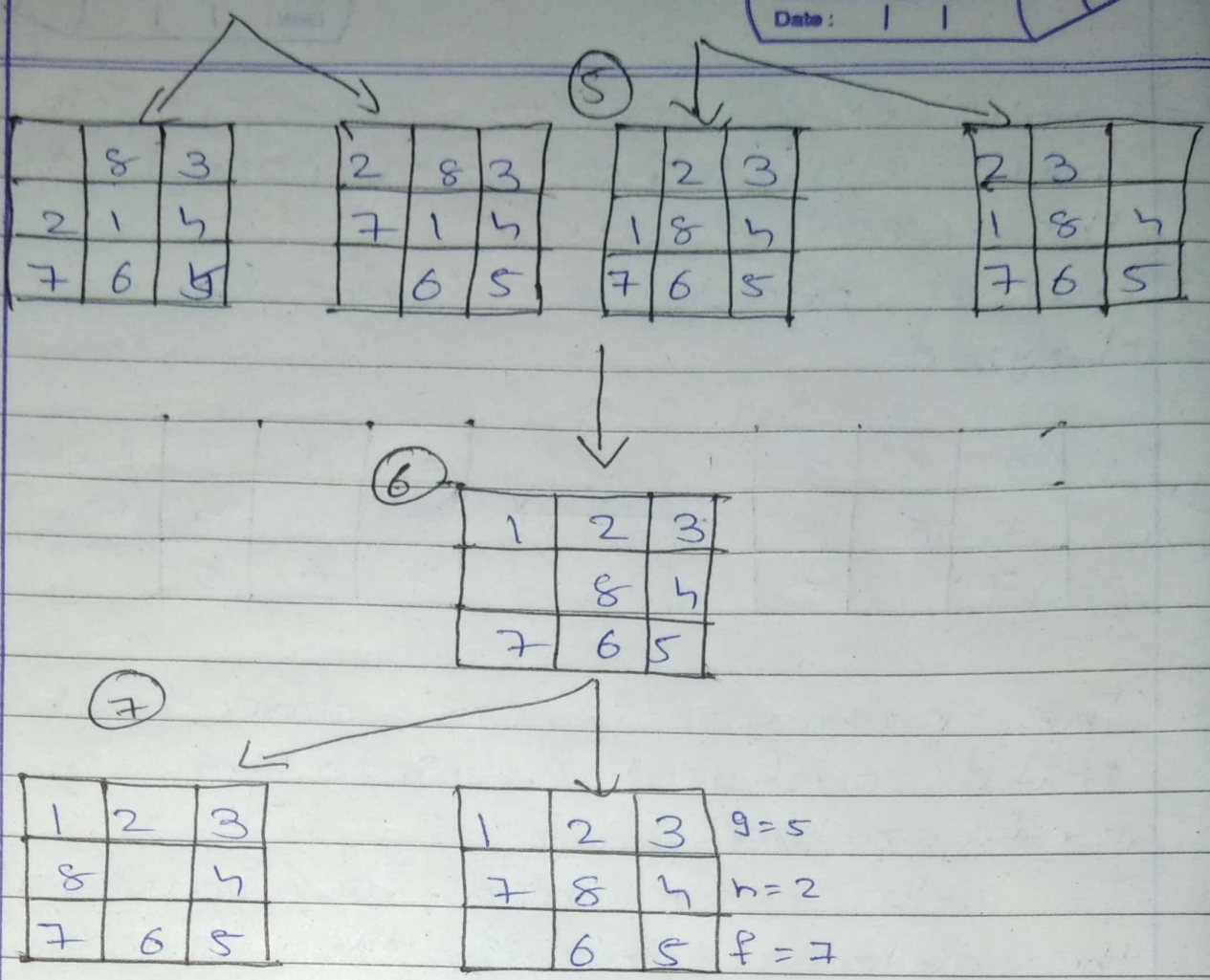
empty thus making it possible to move an adjacent numbered tile into the empty cell.

- Such a puzzle is illustrated in following diagram.



- The program is the change the inital configuration into the goal configuration.
- To solve a problem, we must specify the global database, the rules & the control strategy.





- It has 3 components :-

(i) States (ii) Moves (iii) Goal

- The set of all possible configuration in the problem space, consists of 3,62,880 different configuration of the 8 tiles & blank space.

13. Explain why it is necessary to choose appropriate 'granularity' for knowledge representation.

Ans KR has two distinct entities:
- knowledge is a description of the woold. It determines a system's competence by what it knows
- Representation is the way knowledge is encoded. It defines a system's performance in doing something

* Choosing granularity: At what level of detail should the knowledge be represented?
- Regardless of the KR formalism, it is necessary to known:
  - ↳ At what level should the knowledge be represented & what are the primitives?
  - ↳ Should these be a small numbes of should they be a large numbes of low-level primitives or high-level facts.
  - ↳ High-level facts may not be adequate for inter ference while low-level primitives may require a lot of storage.

- Example of Granularity
- Suppose we are interested in following facts:
  John spotted Sue.

- This could be represented as spotted (agent (John), object (sue))
- Such a representation would make it easy to answer question such as:
  Who spotted sue?
- Suppose we want to know:
  Did John see sue?
- Given one one fact, we cannot discover that answer.
- We can add other facts, such as spotted (x, y) → saw (x, y)
- We can now infer the answer to the question.

14) What is "iterative deepening"? How is it useful in time constrained search?

Ans
- Depth first search is incomplete if there is an infinite branch in the search tree.
- Infinite branches can happen if:
- Paths contain loops
- Infinite number of states and/or operators
- For problems with infinite (or just very large) state spaces, several variants of depth-first search have been developed:
- depth limited search
- Iterative deepening search
- Depth limited search (DLS) is a form of depth-first search.
- It expands the search tree depth-first

up to a maximum depth l.
- The nodes at depth l are treated as if they had no successors.
- If the search reaches a node at depth l whose the path is not a solution, we backtrack to the next choice point at depth $< l$
- Depth-first search can be viewed as a special case of DLS with $l = \infty$
- The depth bound can sometimes be chosen based on knowledge of the problem.
- For e.g., in the route planning problem, the longest route has length $s-1$, where s is the number of cites (states), so we can set $l = s-1$.
- For the most problems, d is unknown.
- Iterative deepening (depth-first) search (IDS) is a form of depth limited search which progressively increases the bound.
- It first tries $l=1$, then $l=2$, then $l=3$, etc. until a solution is found.
- Solution will be found when $l=d$.
- IDDFS combines depth-first search's space-efficiency and breadth first search's fast search.
- (for nodes closes to root)
- IDDFS calls DFS for different depths starting from an initial value. In every call, DFS is restricted from going beyond given depth. So basically we do DFS in a BFS fashion.

15. Explain the algorithm for steepest-Ascent Hill climbing. Briefly describe the situations in which hill climbing may fail to find a solution.

Ans - Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem.
- It terminates when it reaches a peak value where no neighbor has a higher value.
- It is a technique which is used for optimizing the mathematical problems.

- **Steepest-Ascent Hill Climbing:-**

- It is a variation of simple hill-climbing algorithm.
- This algorithm examines all the neighbouring nodes of the current state & selects one neighbour node which is closet to the goal state.
- This algorithm consumes more time as it searches for multiple neighbours

Step-1] Evaluate the initial state, if it is goal state then return success & stop, else make current state as initial state.

step-2] Loop until a solution is found or the current state does not change.

(a) Let target be a state such that any successor of the current state will be better than it.

(b) For each operator that applies to the current state.

(i) Apply the new operator & generate a new state.

(ii) Evaluate the new state

(iii) If it is goal state then return it and quit, else compare it to the target.

(iv) If it is better than target, then set new state as target

(v) If the target is better than the current state, then set current state to target.

step-3] Exit.

- Hill climbing cannot reach the optimal/ best state if it enters any of following regions

(i) Local Maximum (ii) Plateau (iii) Ridge

16b Differentiate with example representation of "Instance" and "Isa" relationships.

Ans - Instance :- It represents the thing that belongs to a class of a particular object.

- The predicate instance is a binary one whose first argument is an object & whose second argument is a class to which object belongs.

- Example :

(i) Marcus is a man
object class
man(Marcus) or instance(marcus, man)

(ii) Marcus was a Pompeian
object class
instance (pompeian, marcus)

(iii) All Pompeian where Romans
$\forall x$: instance (x, pompeian) → instance (x, Roman)

- IS-A relationship :- It represents the class of an object

- Example
(i) Marcus is a man
(ii) Marcus was a pompeian
(iii) All Pompeian were Romans
isa (Pompeian, Roman)

17) Explain with example how choosing the granularity of representation and finding the right structure are crucial issues in knowledge represen-tation?

Ans - KR & Reasoning (KR, KRR) is the part of AI which concerned with AI agents thinking & how thinking contributes to intelligent behaviour of agents

- It is responsible for representing information about the real world so that a computer can understand & can utilize this knowledge to solve the complex real world problems such as diagnosis a medical condition or communicating with humans in natural language

- choosing Granularity:- At what levels of data should the knowledge be represented?
- At what level the knowledge represented & what are the primitives?
- should there be a small number of should there be a large number of low-level primitives or high-level facts.
- High level facts may not be adequate for interference while low-level primitive may require a lot of storage

- Example of granularity.

- Suppose we are interested in following facts John spotted sue.
- This could be represented as spotted (agent (John), object (sue)).
- Such a representation would make it easy to answer questions such are:
- Who spotted sue?
- Suppose we want to know:
  Did John see sue?
- Given only one fact, we cannot discover that answer.
- We can add other facts, such as
  spotted (x, y) -> saw (x, y)

- Finding right structure:- Given a large amount of knowledge stored in a database, how can relevant parts are accessed when they are needs.
- This is about access to right structure for describing a particular situation.
- This requires, selecting an initial structure & then revising the choice.
- While doing so, it is necessary to solve following problem.
- how to perform an initial selection of the most appropriate structure.
- how to fill in appropriate details from

the current situations
- what to do if none of the available structure is appropriate.
- when to create & remember new structure.
- There is no good, general purpose method for solving all these problems. Some KR techniques solve some of these issues.

18) What is wrong with the following arguments?
- Men are widely distributed over the earth
- Socrates is a man
- Therefore, socrates is widely distributed over the earth

How should the facts represented by these sentences be represented in logic so that this problem does not arise?

Ans - The problem is that the first statement is universally quantified while the second statement is only valid for socrates so we have to use universal elimination to be able to derive anything like statement.

- we can represent the facts like this:

1. $\exists y$(place on Earth(y) $\wedge$ Man(f(y)) $\rightarrow$ On(y, f(y))
Man (socrates)

2. $\exists y$ (place on Earth (y) $\wedge$ Man (Socrates) $\rightarrow$ On (y, socrates)

- Where On (y, x) means x is can be found on place y.
- This world express something close to what the first statement says
- For any place or earth, one can find some man there [and therefore, men are widely distributed throughout the earth).

19b What is ment by "control strategy"? State the requirements of a good control strategy.

Ans - It helps us decide which rule to apply next during the process of searching for a solution to a problem
- what to do when there are more than 1 material rules?

- requirements of good control strategy:
(i) cause motion
(ii) Systematic

- Cause motion :- Each rule or strategy applied should cause the motion because if there will be no motion then such control strategy will never lead to a solution

- Motion states about the change of state. if a state will not change then there be no movement from an initial state & we would never solve the problem.

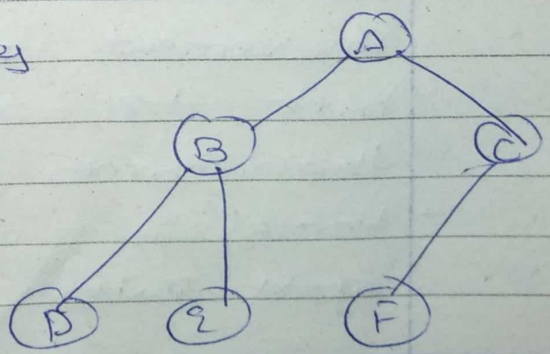
- Systematic :- Through the strategy applied should create the motion but if do not follow some systematic strategy then we are likely to reach the same state number of times before reaching the solution which increases the no. of steps.

- Taking case of only first strategy we may go through particular useless sequences of operats several times

- Control strategy should be systematic implies a need for global motion (over the course of several steps) as well as for local motion [over the course of single step]

- Examples:-

- BFS :- It searches along the breadth & follows FIFO queue data structure
A-B-C-D-E-F



- DFS :- It searches along the depth & follows the stack approach. A-B-D-E-C-F ; it scans all the sub nodes of parent nodes & then moves to another nodes.

20. What is production system? Explain it with an example. Discuss the characteristics of a production system.

Ans - It is also known as production rule system.
- It is a kind of cognitive architectural that is used to implement search algorithms & replicate human problem solving skills.
- It consists of two components
(i) Rule :- It recognize the condition
(ii) Action :- It has the knowledge of how to deal with the condition.

- characteristic :-

- Monotonic Production System :- In this, use of one rule never prevents the involvement of another rule when both the rules are selected at the same time.
- Hence, it enables the system to apply rules simultaneously

- **Partially commutative Production System:-**

In this system if a set of rules is used to change state A to state B then any allowable combination of these rules will also produces the same results (convert state A to state B)

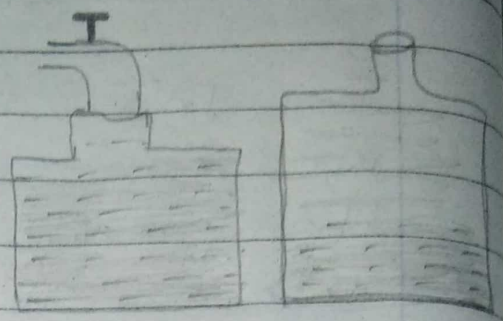
- **Non-monotonic Production System:** It increases the problem solving efficiency of the machine by not keeping a record of the changes made in the previous search process

- It is useful from an implementation point of view as they do not backtrack to the previous state when it is found that an incorrect path was followed.

- **Commutative Production System:** These type of production system is used when the order of operation is not important & the changes are reversible

- **Example:-** We have two jugs of capacity 5L & 3L, & a tap with an endless supply of water. The objective is to obtain 4L exactly in the 5L jug with the minimum steps possible

- **Production system :**



- Fill the 5L jug from top
- Empty the 5L jug
- Fill the 3 litres jug from tap
- Empty the 3L jug
- Then empty 3L jug to 5L
- Empty the 5L jug to 3L
- Pour water from 3L to 5L
- Pour water from 5L to 3L but do not empty

Solution:- 1, 8, 4, 6, 1, 8 or 3, 5, 3, 7, 2, 5, 3, 5

21. What do you mean by admissibility of an algorithm? Is A* algorithm an admissible one? when?

Ans - If the heuristic function is admissible, it means that it never overestimate the actual cost to get to the goal, A* is guaranteed to return a ~~tree~~ cost-cost path from start to goal.

- A search algorithm is admissible if it is guaranteed to find a minimal path to a solution whenever such a solution exists.

- BFS is admissible, because it looks at

every state at level n before considering any state at level n+1.

- If the true minimum cost of getting from node n to a goal state is c then h must satisfy : $h'(n) \leq c$

- If h' is a perfect estimator of h, then A* will converge immediately to the goal state with no search.

- If h' never overestimates h, then A* algorithm is guaranteed to find an optimal path if one exists

- h(n) is admissible if $\forall n\ (h(n)) \leq h^*(n)$ where h(n) = cost indicated by h to reach a goal from n
$h^*(n)$ = is the optimal cost to reach a goal from n.

- If an admissible heuristic is med, A* search algorithm will never return a subpptimal goal node.