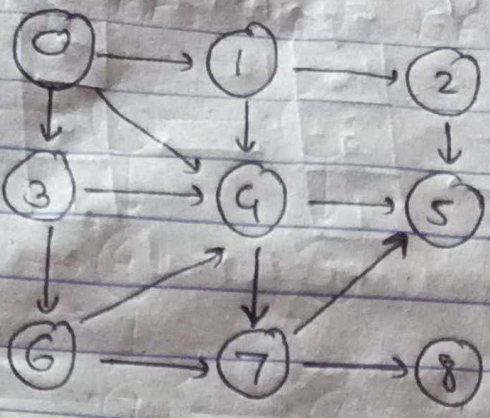
Q.1 Solve using BFS and DFS, start from 0 node:-



BFS :- [0] → { 0 is starting node and have three child after removal of zero from queue his child are inserted into queue }

[0 | 1 | 4 | 3]

[4 | 3 | 2]

[3 | 2 | 5 | 7]

[2 | 5 | 7 | 6] { 2 have child 5 already present in the queue }

[5 | 7 | 6]

[7 | 6]

[6 | 8]

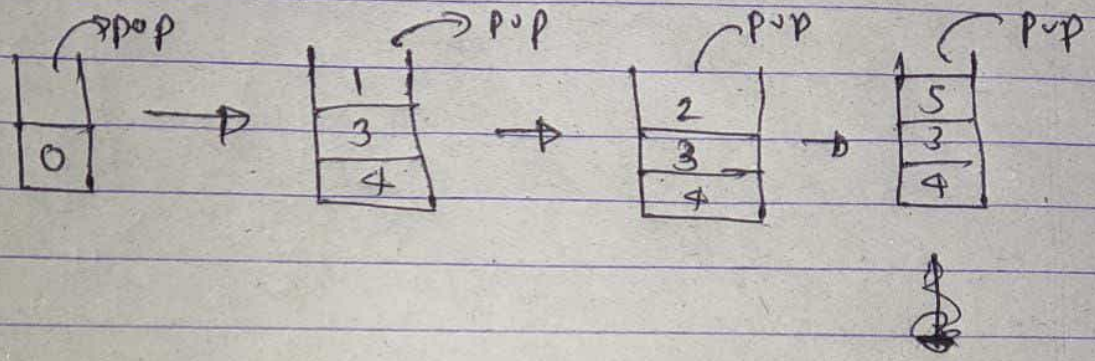
[8]

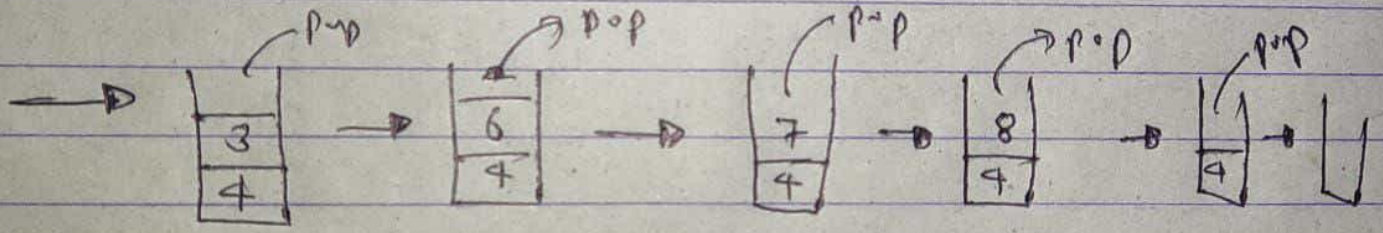
[ ]

Now, total graph is visited or all node are visited.

0 → 1 → 4 → 3 → 2 → 5 → 7 → 6 → 8

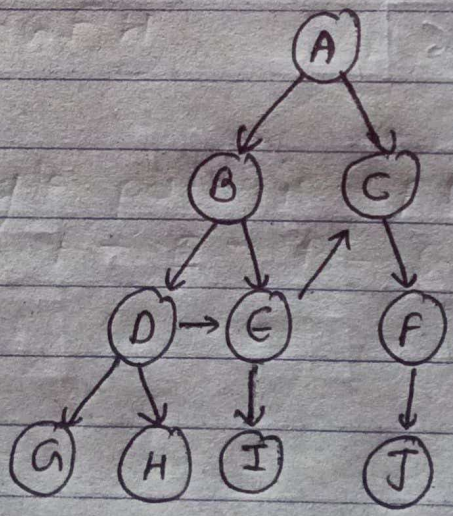
DFS

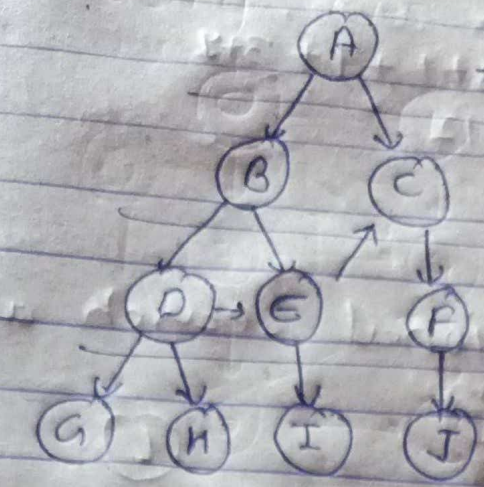




0 → 1 → 2 → 5 → 3 → 6 → 7 → 8 → 4

Q.2 Solve using depth limited search algo where $d = 2$.

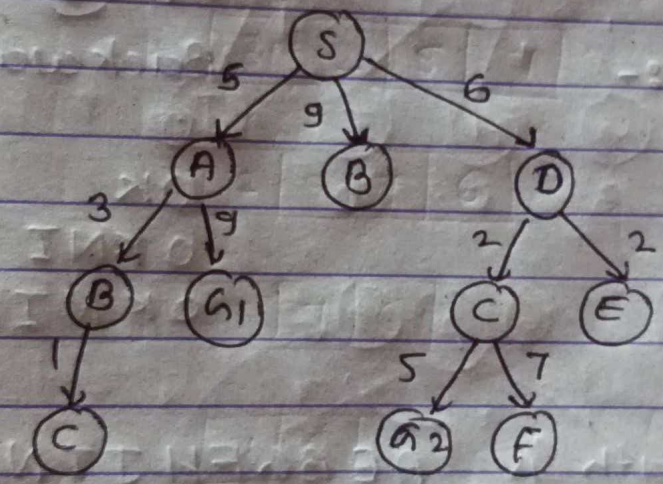




Here $d = 2$

Path :- $A \rightarrow B \rightarrow D \rightarrow E \rightarrow C \rightarrow F$

Q.4 solve the uniform cost search.



$S \xrightarrow{5} A$ ✓ (min path length)

$S \xrightarrow{9} B$

$S \xrightarrow{6} D$

then

$A \xrightarrow{3} B$ ✓ (min path length)

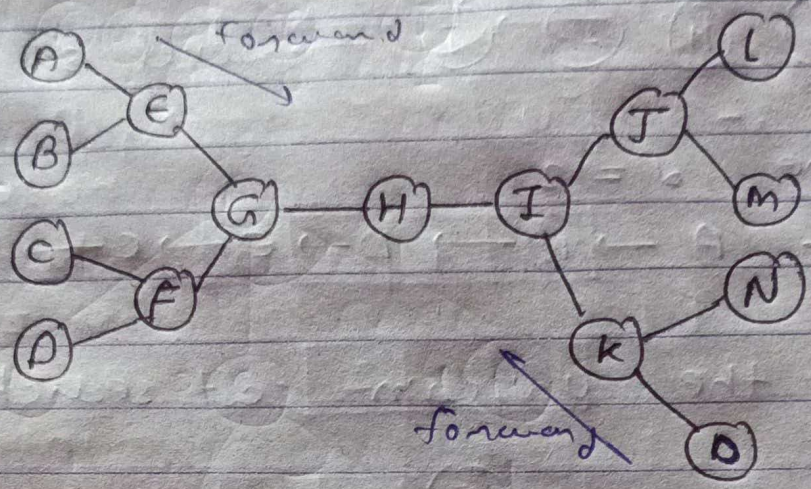
$A \xrightarrow{9} G_1$

$B \xrightarrow{1} C$ (goal state not find perform backtracking

$S \rightarrow A \rightarrow B \rightarrow C \rightarrow B \rightarrow A \rightarrow G_1$

Total cost = 5+3+1+1+3+9
= 10+12
= 22

Q.5 solve using bidirectional search algo.



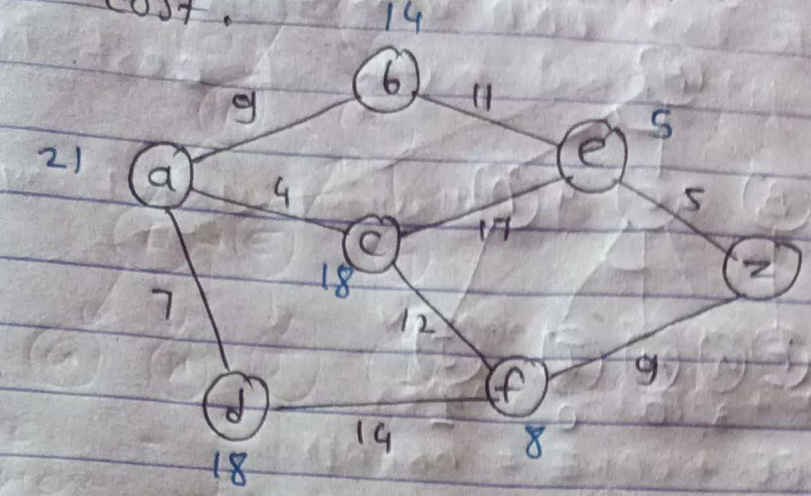
⇒

forward :-	Backward :-
A	O
AE	OK
AEBG	OKNI
AEBGF(H)	OKNIJ(H)

single path is AEBGFHJINKO

∴ A → E → B → G → F → H → J → I → N → K → O

Q.7 solve using A* algo and find best path along with cost.



⇒ Goal state = z

Total cost = path cost + Heuristic value

step (1) :-
a → b = 9+14 = 23 Hold
a → c = 4+18 = 22 ← (min)
a → d = 7+18 = 25 Hold

step (2) :-
a → c → e = 4+17+5 = 26 Hold
a → c → f = 4+12+8 = 24 ←

step (3) :- a → c → f → z = 4+12+9+0 = 25 — (R1)

~~step (4)~~ :- a → c → e → z = 4+17+5+0 = 26 — (R2)

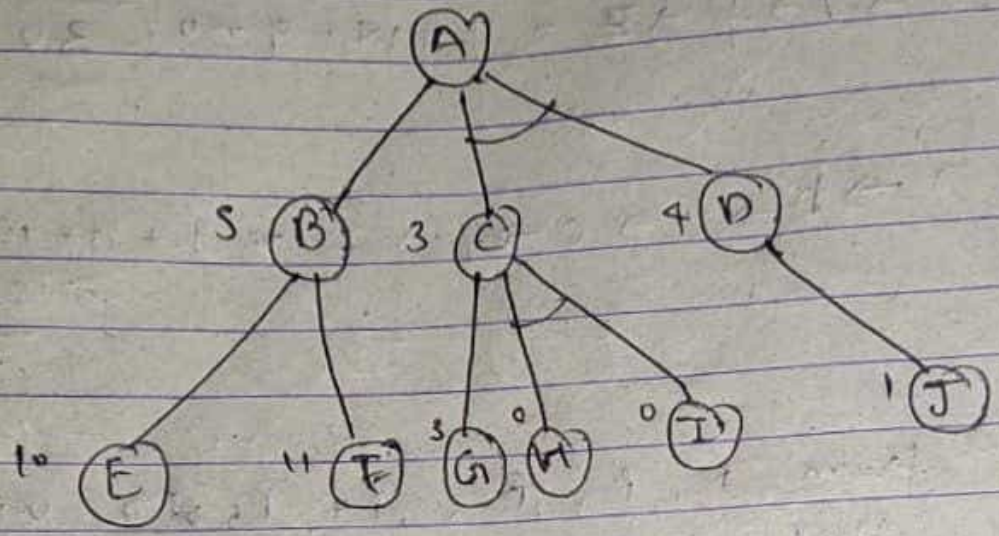
step (5) :- a → b → e → z = 9+11+5 = 25 — (R3)

a → d → f → z = 7+14+9 = 30 — (R4)

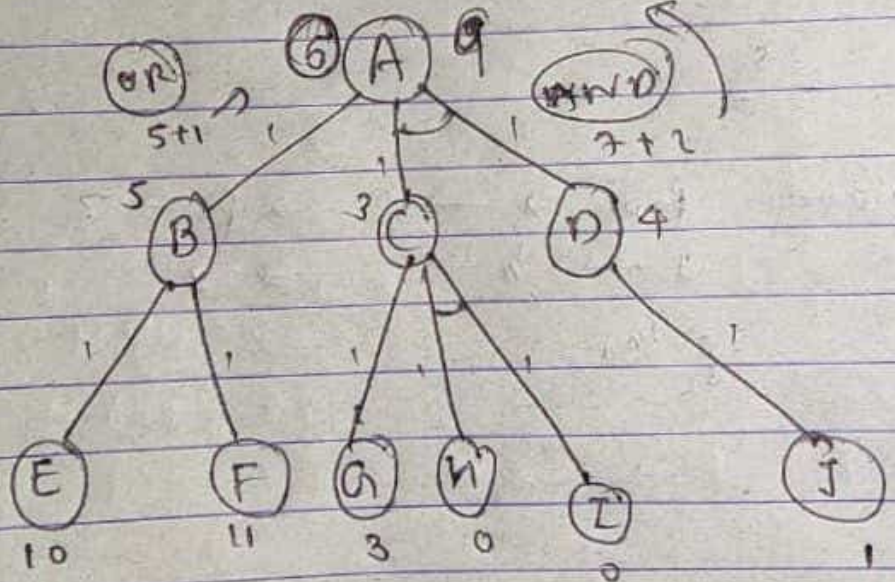
from R1, R2, R3, R4 we get that R1 & R2 are the shortest path to choose.

cost = 25 ∴ Best path is R1, R3

(8) AO*



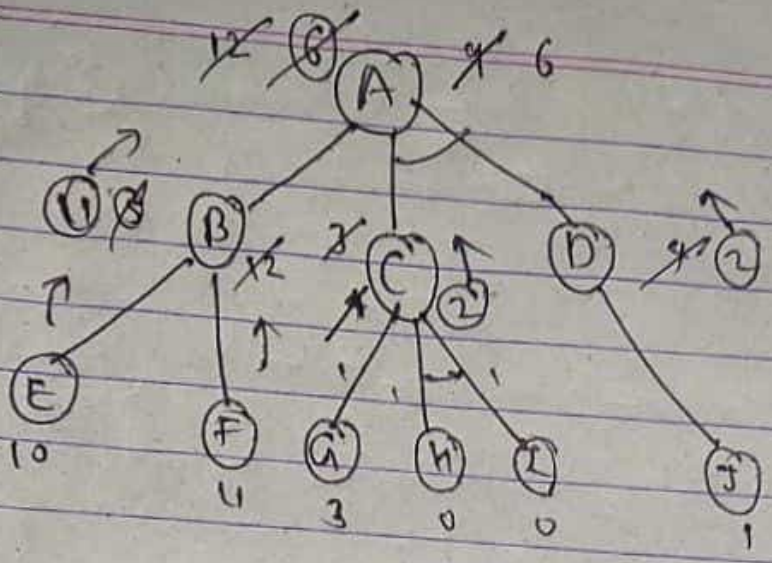
→ firstly go toward OR then AND.



Here OR → (6) A → B

AND → 9 A → C & D

select (6) OR



At B ⇒ B → E = 1 + 10 = 11

B → F = 1 + 11 = 12

Then A → B → E = 1 + 1 + 10 = 12

But 12 > 9 so go for AND direction.

At C ⇒ there is OR & AND

C → G ⇒ 1 + 3 = (4)

C → H & I ⇒ 1 + 1 + 0 + 0 = (2)

select (2) at C

At D ⇒ D → J = 1 + 1 = (2)

Now At A ⇒ A → C & D

⇒ 1 + 1 + 2 + 2 = (6)

6 < 9

So in AND we got final path & Cost is 6 which is

Q.18 Provide a soln of following using cryptarithm puzzle,

1. SEND + MORE = MONEY
2. BASE + BALL = GAMES
3. COFFEE + CAKE = BREAD
4. PANDA + PANDA = ANIMAL
5. CAT + DOG = BIRD
6. SHEEP + PIP = FARM

⇒ (1) send + more = money

S	E	N	D
9	5	6	7

+

M	O	R	E
1	0	8	5

M	O	N	E	Y
1	0	6	5	2

```
 9567
+1085
------
10652
```

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

∴ value of SEND = 9567,
MORE = 1085,
MONEY = 10652

(2) Base + Ball = Games

B	A	S	E
7	4	8	3

B	A	L	L
7	4	5	5

G	A	M	E	S
1	4	9	3	8

```
  7 4 8 3
+ 7 4 5 5
---------
1 4 9 3 8
```

here, B → 7
A → 4
S → 8
E → 3
L → 5
G → 1
M → 9

m → g

Q.9 Describe the basic idea of the hill-climbing search algo.

Q.10 limitation of hill-climbing algo

Q.12 advatage & disadvantages of hill climbing

Q.11 Types of hill climbling algo.

→ (1) Hill climbing algo is local search algorithm which continuously moves in the direction of increasing 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.

(2) Hill climbing algorithm is a technique which is used for optimizing the mathematical problems. one of the widely discussed examples of Hill climbing algo is traveling-salesman problem in which we need to minimize the distance traveled by the saleman.

(3) It is also called greedy local search as it only looks to its good immediate neighbor state and not beyond that.

(4) A node of hill climbing algo has two components which are state & value.

(5) Hill climbing is mostly used when a good heuristic is available.

- local search algo
- signal path / directed
- No backtracking

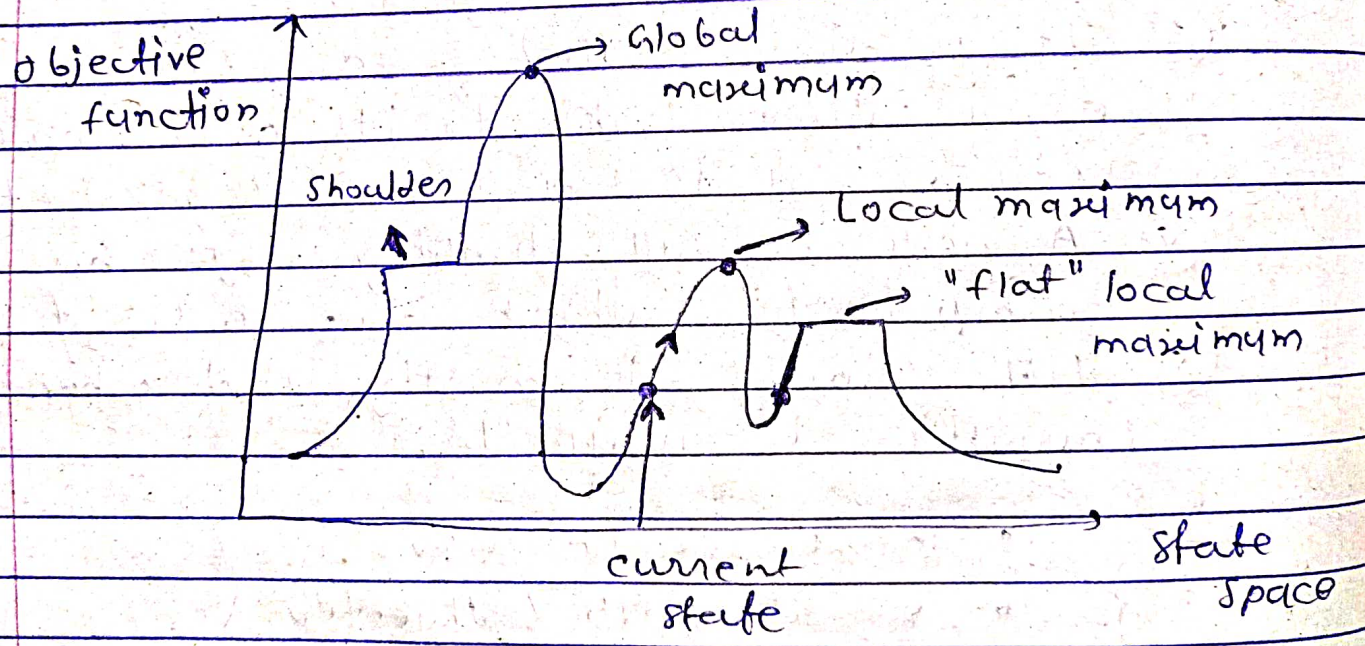
Page No.
Date

feature of hill climbing :-

- Generate of & test variant
- Greedy approach
- No backtracking

state - space diagram for Hill - climbing :-

The state - space landscape is a graphical representation of the hill - climbing algo on y-axis we have taken cost function, state - space on the x-axis.



Local maximum :-

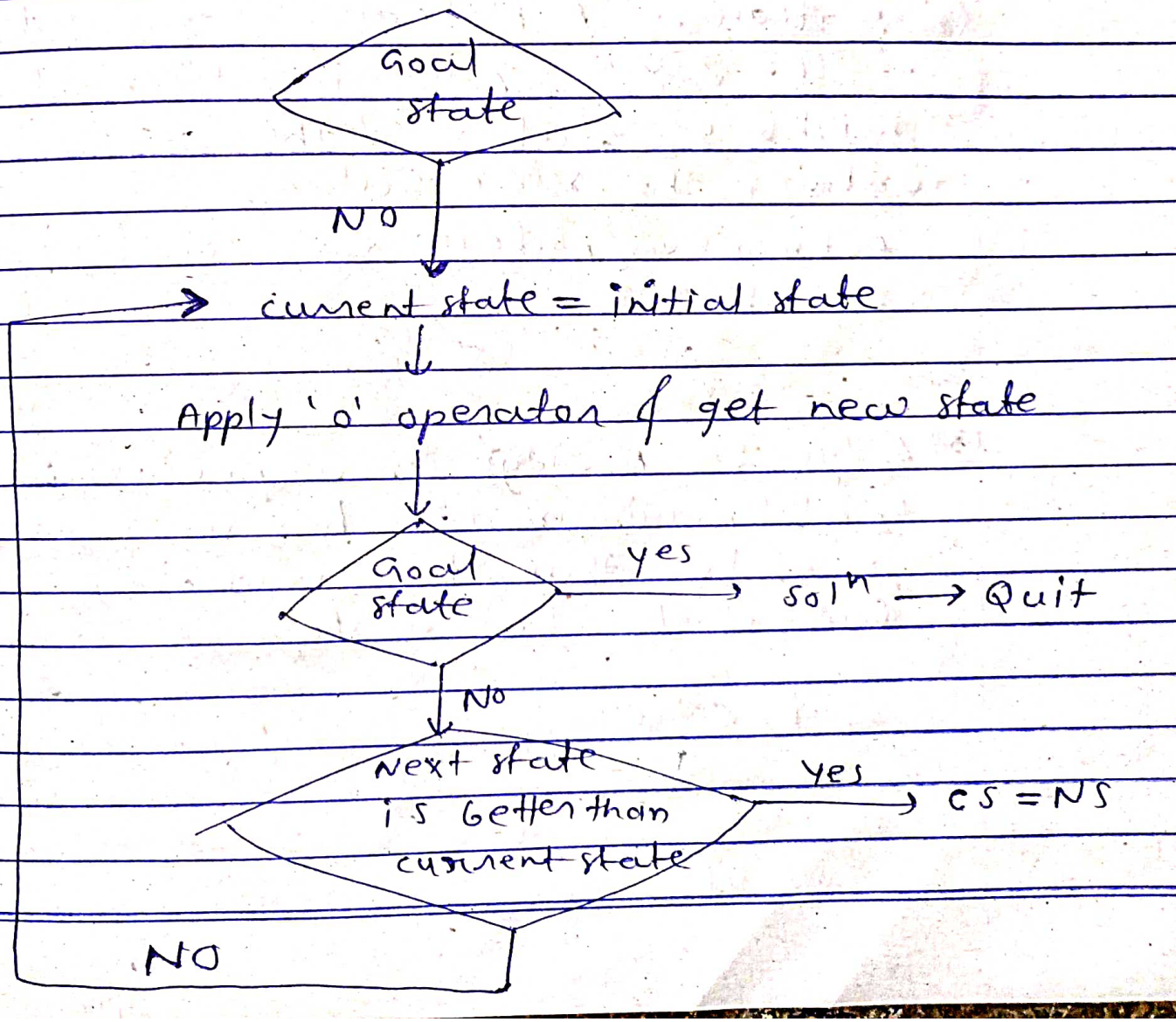
local maximum is a state which is better than its neighbour states but there is also another state which is higher than it

Global maximum :- It is the best possible state of state space landscape. It has the highest value of objective fn.

Flate local maximum :- where all the neighbour states of current state have the same value.

Shoulder :- It is a plateau region which has an uphill edge.

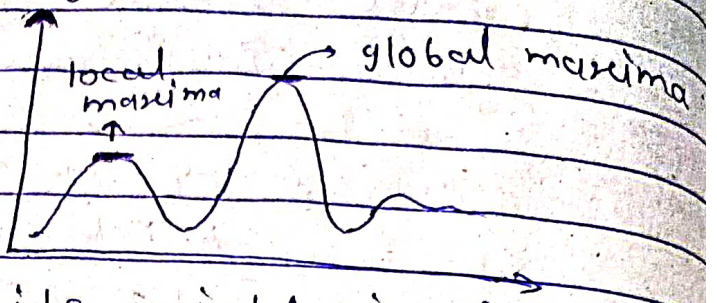
flowchart :-



Page No.
Date

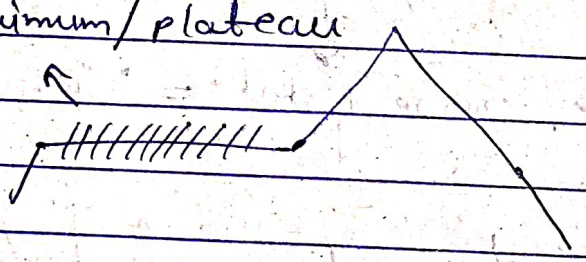
Limitation (disadvantage) :-

(1) Local maxima :- A local maximum is a peak state in the landscape which is better than each of its neighboring states, but there is another state also present which is higher than the local maximum.

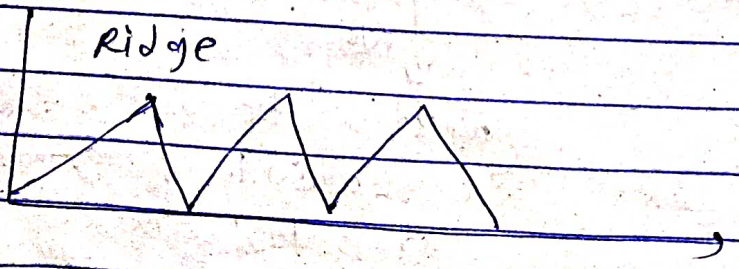


soln :- Backtracking technique

(2) Plateau :- A plateau is the flat area of the search space in which all the neighbours states of the current state contains the same value.



(3) Ridge :- A ridge is a special form of the local maximum. It has an area which is higher than its surrounding areas, but itself has a slope



Page No.
Date

Advantage :-

- simplicity
- Efficiency in local search spaces
- memory efficiency
- low computation cost.

Types of Hill climbing algo :-

① simple Hill climbing :-

simple hill climbing is the simplest way to implement a hill climbing algo. It only evaluates the neighbor node state at a time & selects the first one which optimize current cost and set it as a current ~~state~~ state.

feature :-

- less time consuming
- Less optimal solution and the soln is not guarantee.

Algo :-

(follow flow chart & write algo according to that)

② Steepest - Ascent Hill climbing :-

The steepest - Ascent algo is a variation of simple hill climbing algo. This algo examines all the neighboring nodes of the current state and selects one neighbor node which is closet to the goal state.

algo :-

(3) Stochastic hill climbing :-
Stochastic hill climbing does not examine for all its neighbor before moving. Rather, this search algorithm selects one neighbor node at random & decide whether to choose it as a current state or examine another state.

Q.13 Give an example of a problem where Best-first search is a suitable choice.

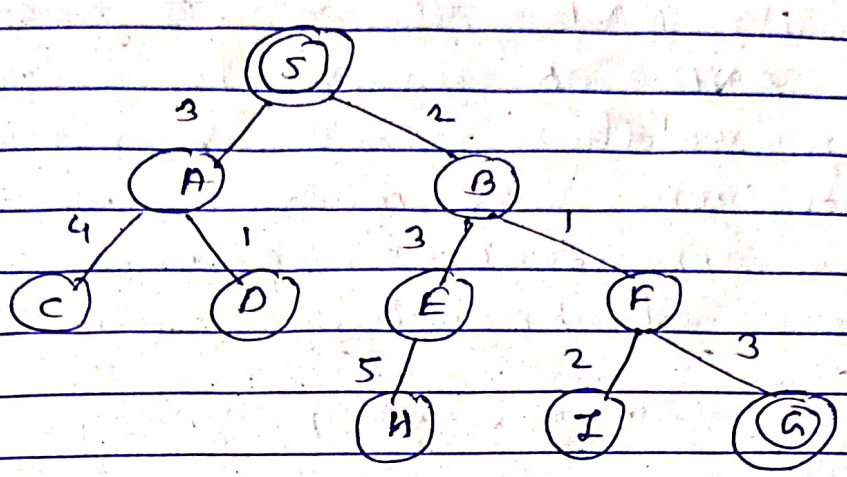
→ (1) Greedy best-first search algo always selects the path which appears best at that moment.
(2) It is the combination of DEpth-first-search and & BFS algo.
(3) It uses heuristic function of search.
(4) best-first search allows us to take the advantage of both algo.
(5) with the help of best-first search at each step we can choose the most promissing node.
(6) It is implemented by priority queue.

advantage :-
- Best first search can switch between BFS & DFS by gaining the advantage of both the algo.
- This algo is more efficient than BFS & DFS algo.

Disadvantage :-

- It can behave as an unguided DFS in the worst case scenario.
- It can get stuck in a loop as DFS.

ex



open list	closed list
[S]	[ ]
[B A]	[S]
[A]	[S B]
[F E A]	[S B]
[E A]	[S B F]
[G I E A]	[S B F]
[G I E A]	[S B F G]

S → B → F → G

Page No.
Date

Q.14 What is an AND-OR graph and how does it represent knowledge?

Q.16 What is significance of backtracking in AND-OR graph traversal?

→ An AND-OR graph is a graphical representation used in artificial intelligence to model complex relationships and dependencies between different pieces of knowledge or propositions.

It is directed graph where nodes represent logical propositions or conditions and edge represent relationships between these propositions.

Representation in an AND-OR Graph:-

① AND nodes :-

- An AND node represents a logical conjunction, meaning that all its child nodes must be true for the parent node to be true.
- Children of an AND node are connected by edges and they collectively contributes to the truth value of the parent AND node.

(2) OR Nodes :-

- An OR node represents a logical disjunction meaning that at least one of its child nodes must be true for the ~~the~~ parents node to be true.
- Children of an OR node are connected by edges, and any one or more of them can contribute to the truth value of the parent OR node.

Q.16

→ significance of backtracking in AND-OR graph traversal lies in its ability to handle choices and decisions within the graph effictively.

(1) handling choices in OR nodes :-

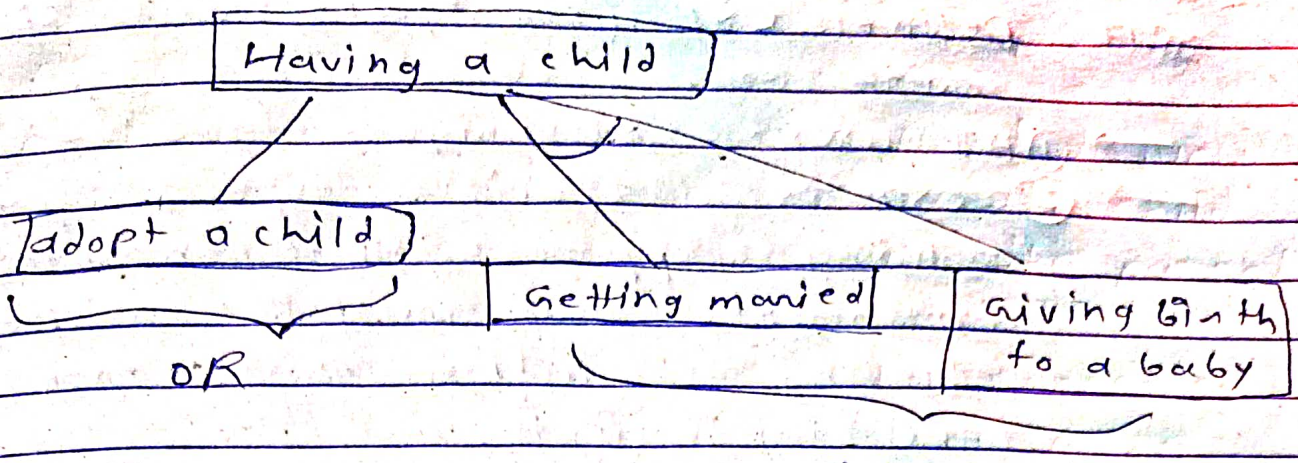
In an AND-OR graph, OR nodes represent points where the system has multiple choices or alternatives.
Backtracking allows exploration of different branches at OR nodes, trying on choice & then backtracking to explore others if the chosen path does not lead to a solution.

(2) Recovery from dead ends :-
When traversing an AND - OR graph, backtracking is essential for recovering from dead ends or situations where a certain choice does not lead to a soln.

It allows the algorithm to go back to the previous decision point (node and try alternative paths, effectively exploring the entire soln space.

Q.15 What is the Ao* algo and how does it differs from the traditional A* algo.

→ Ao* method divides any given difficult problem into a smaller group of problem that are then resolved using the AND - OR graph concept. AND OR graph are specialized graph that are used in problems that can be divided into smaller problem.
The AND side of the graph represents a set of tasks that must be completed to achieve the main goal, while the OR side of the graph represents different methods for accomplishing the same main goal.



working AO* algo :-

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

- actual cost of traversal
- cost from initial node to current node
- estimated cost from the current node to the goal node.

Difference between the A* & AO* :-

- A* algorithm & AO* algo both works on the best first search.
- They are both informed search and work on given heuristics values.
- A* always gives the optimal solution but AO* doesn't guarantee to give the optimal solution.
- once AO* got a soln doesn't explore all possible path A* explore all paths.
- when compared to the A* algo, the AO* algo uses less memory
- opposite to the A* algo, the AO* algo cannot go into an endless loop.

Q.17 Define CSP (constraint satisfaction problem) with ex.

→ (1) AI uses CSP technique to solve complex problem.
(2) CSP is a specific type of problem-solving approach that involves identifying constraints that must be satisfied and finding a solution that satisfies all the constraints.
(3) Finding the solution that meets a set of constraints is the goal of CSP.
(4) There are mainly three basic component is the constraint satisfaction problem:-

- variable
- Domain
- constraints

variables :- The things that need to be determined are variables. variable in a CSP are the objects that must have values assigned to them in order to satisfy a particular set of constraints.
variable can be boolean, integer, categorical variable.

Domains :- The range of potential values that a variable can have is represented by domains.
In sudoku, the set of number from 1 to 9 can serve as the domains of a variable.

constraints :- The guidlines that control how variable ~~value~~ ~~for~~ relate to one another are known as constraints.

Types of constaints :-

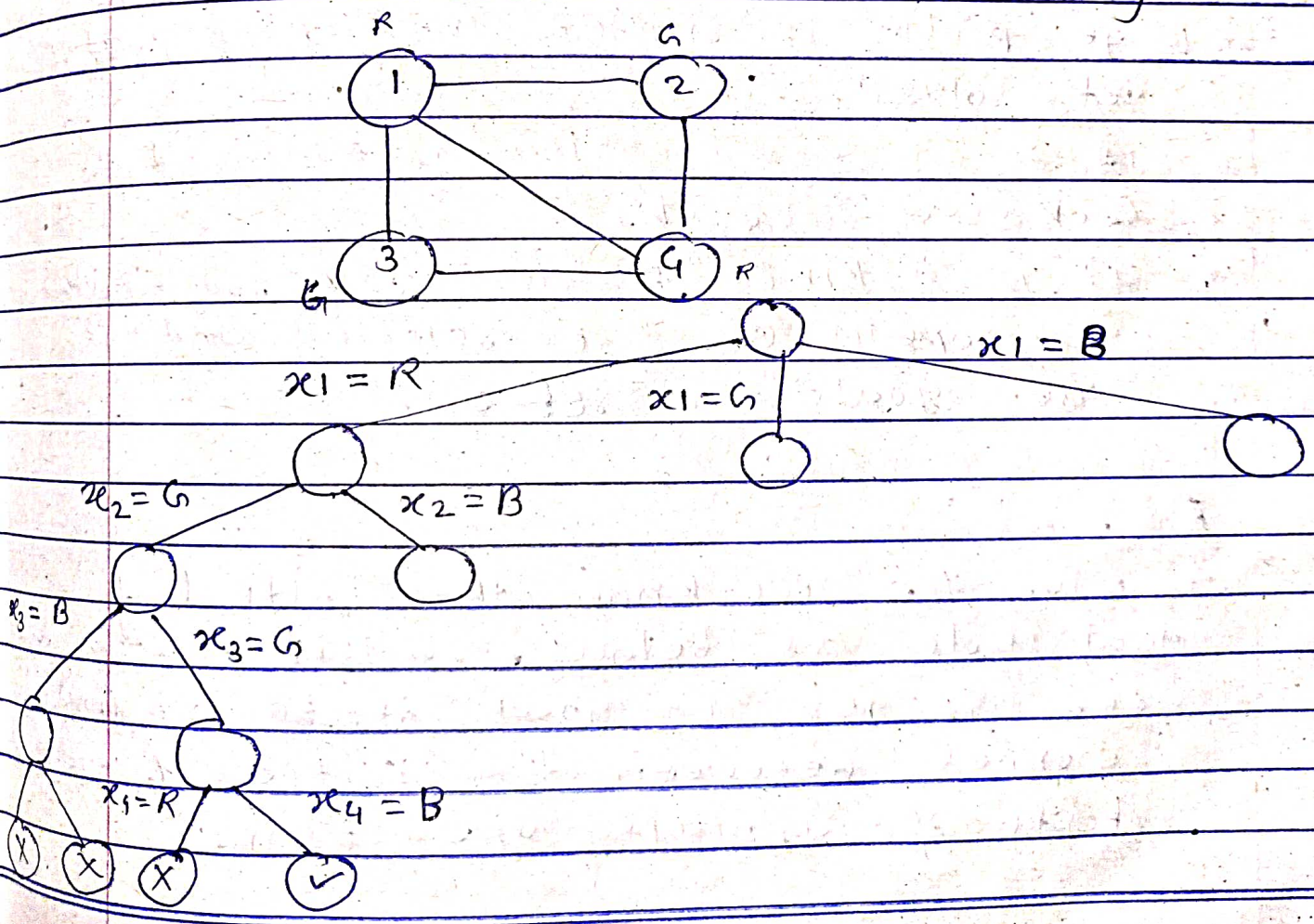
- unary constraint :-
- binary —11—
- Global —11—

ex Graph coloring / node coloring

variable = {1, 2, 3, 4}

Domain = {Red, Green, blue}

constraints = {adjacent node should not have same color}



Page No.
Date

Q.20 mean-end analysis can be used to solve real-world problem

Q.19 What is mean-end analysis & how does it relate to problem solving?

→ we have studied the strategies which can reason either in forward or backward but a mixture of two direction is appropriate for solving a complex & large problem. Such a mixed strategy make it possible that first solve the major part of a problem & then go back & solve the small problems arise during combining the big parts of the problem. Such technique is called mean-end analysis.

- large proble is divided into small part & get solved.
- It is a combination of forward & backward search
- It is a kind off sub-goal mechanism where operaters are selected and then sub-goals are ~~set~~ up.

Ex.

~~To solve the~~ we know the initial & goal state as below. In this problem we needs to get the goal state by finding differences between the initial & goal state & applying operators.

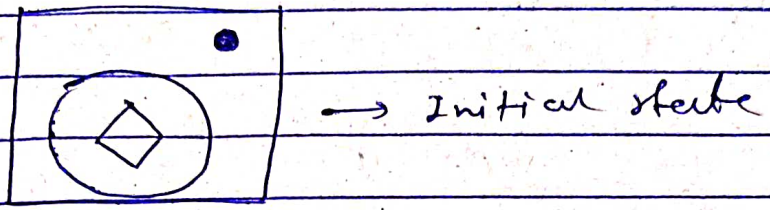
Page No.
Date

Soln :-

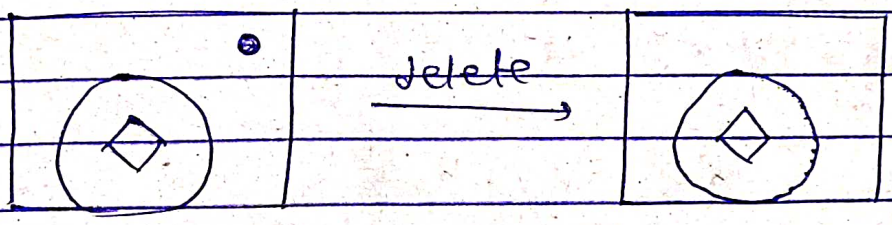
The operators we have for this problem are:

- move
- delete
- expand

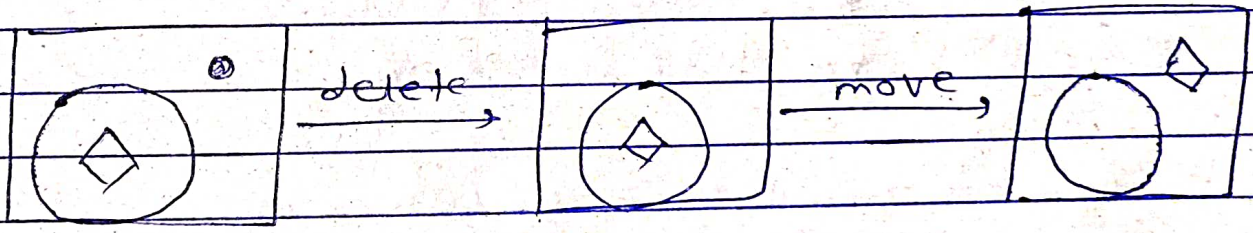
Step (1) Evaluating the intial state :-



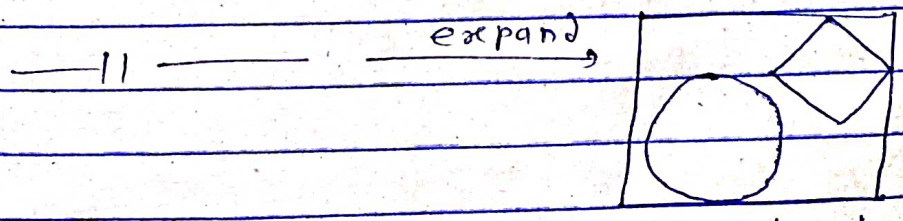
Step (2) Apply delete operaton :-



Step (3) :- Applying move operaton



Step (4) :- Applying expand operaton :-



goal state.

Page No.
Date

another ex. (real world example)
want to become AI/Web developer then we have to go step by steps. like learning html, python, css, Javascript, ml, libraries, etc.

Q.6 What is a heuristic function in the context of heuristic - based search algorithm?

→ Heuristic f^n :- Heuristic is a f^n which is used in informed search, and it finds the most promising path. It takes the current state of the agent as its input and produces the estimation of how close agent is from the goal.

↙ The heuristic method, however, might not always give the best solution, but it guaranteed to find a good solution in reasonable time.

° The values of the heuristic function is always +ve.

$$h(n) <= h^*(n)$$

here $h(n)$ is heuristic cost, & $h^*(n)$ is the estimated cost. Hence heuristic cost should be less than or equal to the estimated cost.