Artificial Intelligence

1/8/24

Python 3.11.x

import. math

$2\pi r$

x^n

math.pow (x,n)

Artificial Intelligence

Artificial

Not natural/Creative

Intelligence

Knowledge

Ability to learn/
Ability to acquire and apply knowledge

Learning → Gaining knowledge

Knowledge → Something which can help to take decision.

Books:

1. Russell & Norvig
2. Ritch & Knight

- **Artificial Intelligence:**

AI is a study how to make computers/devices that behave like human being intelligently. — McCarthy 1956

Syllabus:

Agent
Searching Algo
Game playing
Expert system

Syllabus:

Knowledge representation

- Propositional Calculus
- Fuzzy

6/8/24

Categories of AI

1. Thinking Humanly
2. Thinking Rationaly
3. Acting Humanly
4. Acting Rationaly

AI

Device (Artificial) Intelligence

1. Thinking Humanly – "Machine with minds"
2. Thinking Rationaly – "Right Thing."
3. Acting Humanly – Turing Test
4. Acting Rationaly – "Do right things at right time."

"ALVINN"

"Deep Blue"

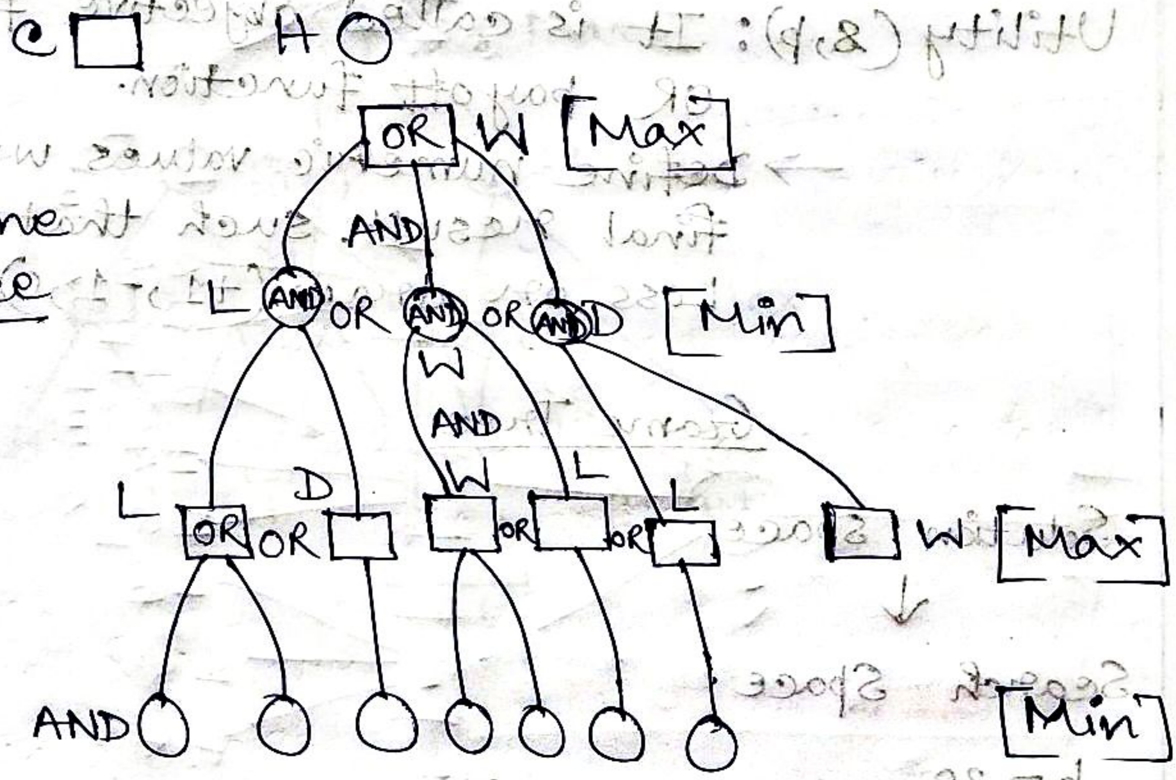
Expert System:–

"FunAsso"
"Dendron"
Mycin

Human Intelligence Vs AI

Game Playing   AO* Algorithm

Deterministic
Turn Taking
Two Players
Zero Sums
Perfect Information Game



Game Tree

MinMax Tree

1. Why Game Tree is Known as MinMax Tree?
2. "MinMax Tree is also Known as ANDOR Tree" — Justify.

→ Max node ≈ OR node
Min node ≈ AND node

S0:- Initial State

Player(s): Define which player has move in the State s

ACTION(s): Return the set of legal moves in State s

RESULT(s,a): Result after state s with score a

Terminal Test(s): Which state define the game is over.

Utility(s,p): It is called objective function OR payoff function.

→ Define numeric values with final result such that win, Loss or draw (+1, -1, 0)

8|8|24

Game Tree

Solution Space

↓

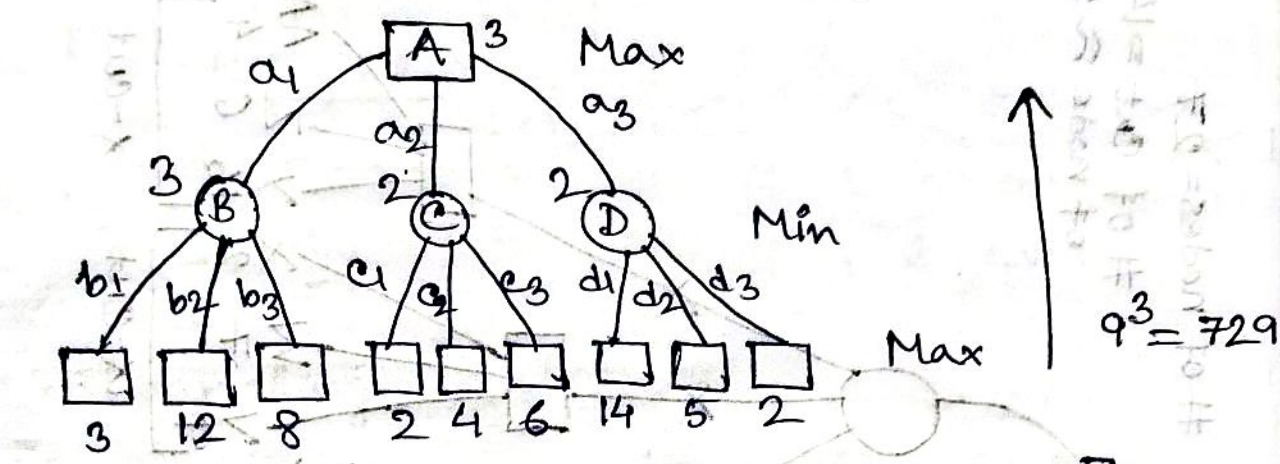
Search Space

$b = 30$

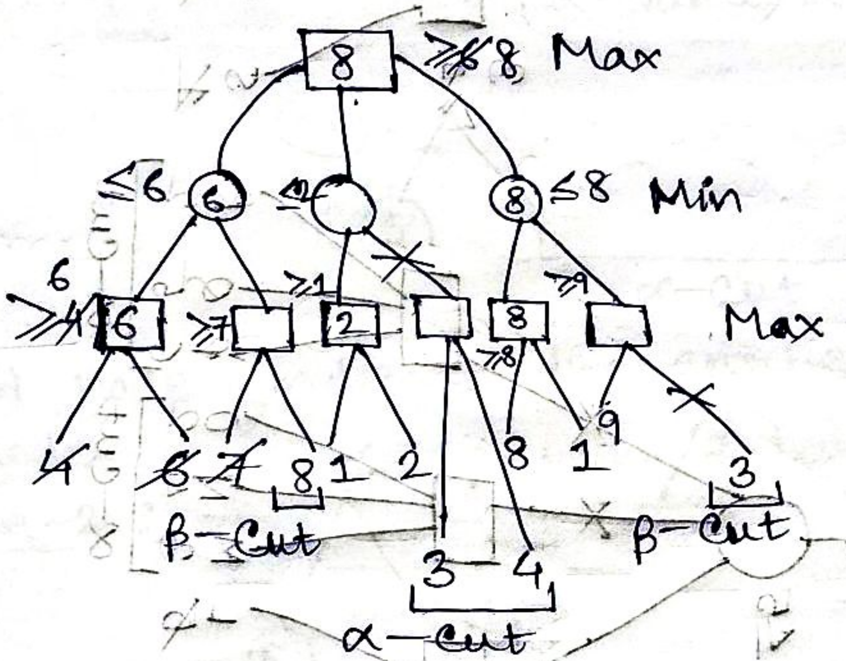
$d = 100$

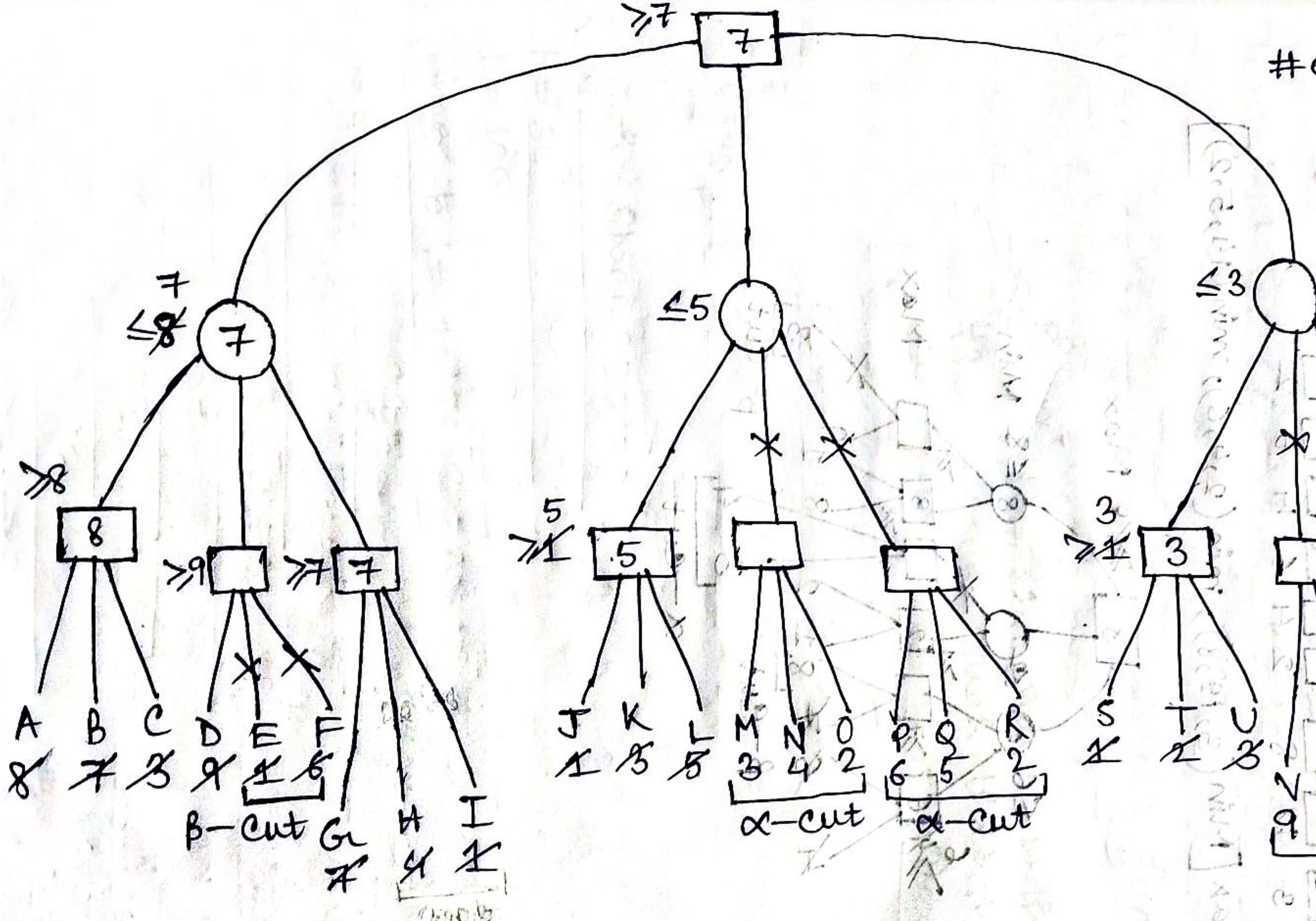
Time $b^d = 30^{100}$

Two Persons Zero Sum Game



Max[Min(3,12,8), min(2,4,6), min(14,5,2)]





# of nodes = 27
# of cut = 14
of node ≈ 50%

Minimum check no. of nodes E is given by

$$E = b^{\lfloor d/2 \rfloor} + b^{\lceil d/2 \rceil} - 1$$

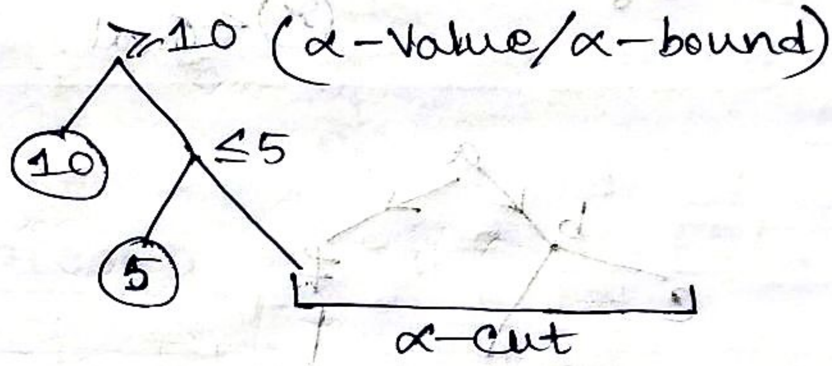
$$= b^{\frac{d-1}{2}} + b^{\frac{d+1}{2}} - 1 \quad \text{if } d \text{ is odd}$$

$$= 2b^{d/2} - 1 \quad \text{if } d \text{ is even}$$

b = Branching Factor

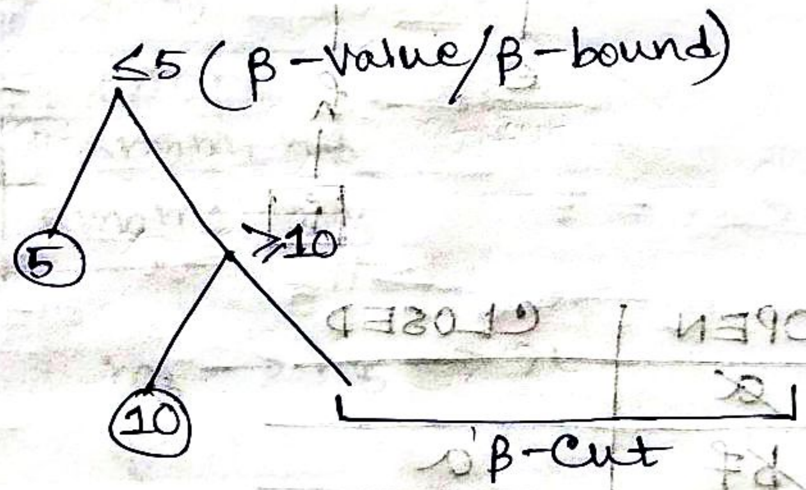
- Observations:

- Case-1:



Bound node indicate the current value of a node.

- Case-2:



- Case-1:

A max ancestor of a min node cut off sons of the min node if current value of min node less than or equals to current value of max node.

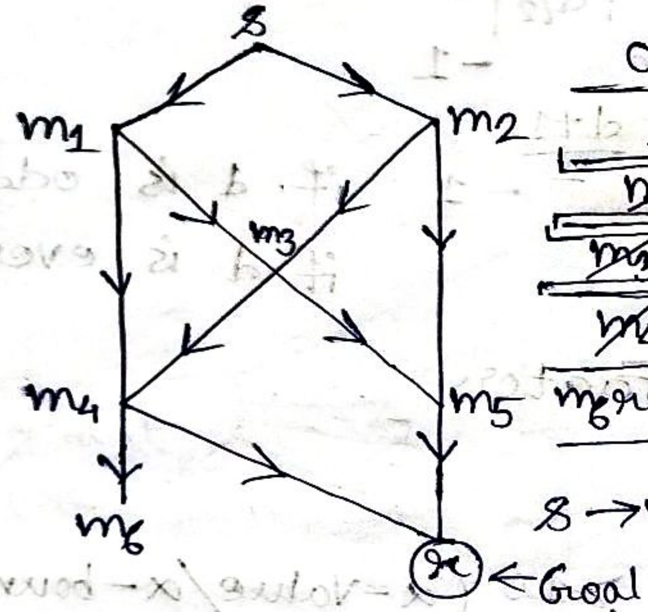
- Case-2:

A min ancestor of a max node cut off sons of max node if current value of max node is greater than or equals to current value of min node.

Searching

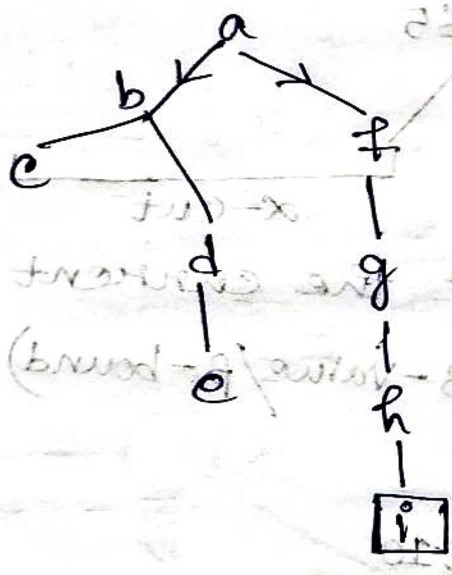
DFS

LIFO (stack)



OPEN	CLOSED
~~s~~	
~~m_1~~m_2	s
~~m_3~~m_4m_2	sm_1
~~m_4~~m_5m_2	sm_1m_3
m_6 r m_5m_2	s$m_1m_3m_4$

$s \to m_1 \to m_3 \to m_4 \to r$



OPEN	CLOSED
~~a~~	
~~b~~f	a
~~c~~df	ab [CLEAN UP]
~~d~~f	ab
~~e~~f	abd [CLEAN UP]
~~f~~	a
~~g~~	af
~~h~~	afg
i	afgh

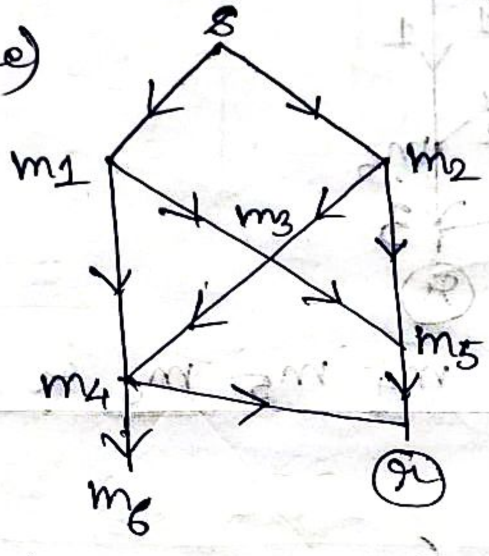
$a \to f \to g \to h \to i$

Goal is not present in the left sub tree

CLEAN UP operation will be performed on the leaf node (Not a goal node) and the ancestors will be cleaned from closed.

BFS

FIFO (Queue)



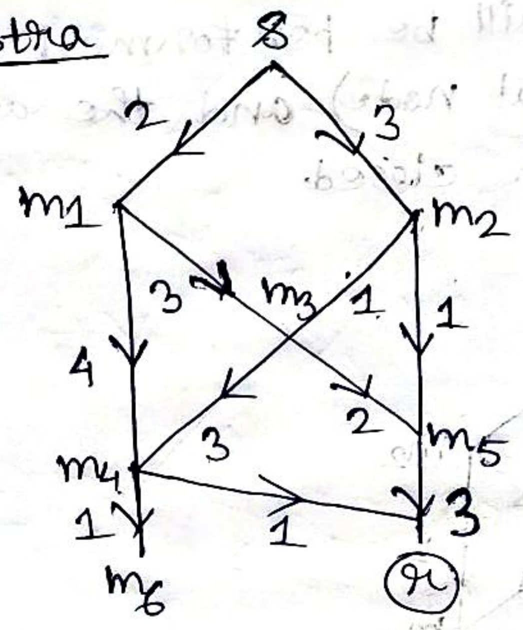
OPEN	CLOSED
~~S~~	
m_1 ~~m_2~~	S
$m_5 m_3$ ~~m_4~~	$S m_2$
$m_4 m_5$ ~~m_3~~	$S m_2 m_1$
m_4 ~~m_5~~	$S m_2 m_1 m_3$
$r m_4$	$S m_2 m_1 m_3 m_5$

$m_3 \rightarrow m_5 \rightarrow r$

$\downarrow$

$m_1 \rightarrow S$

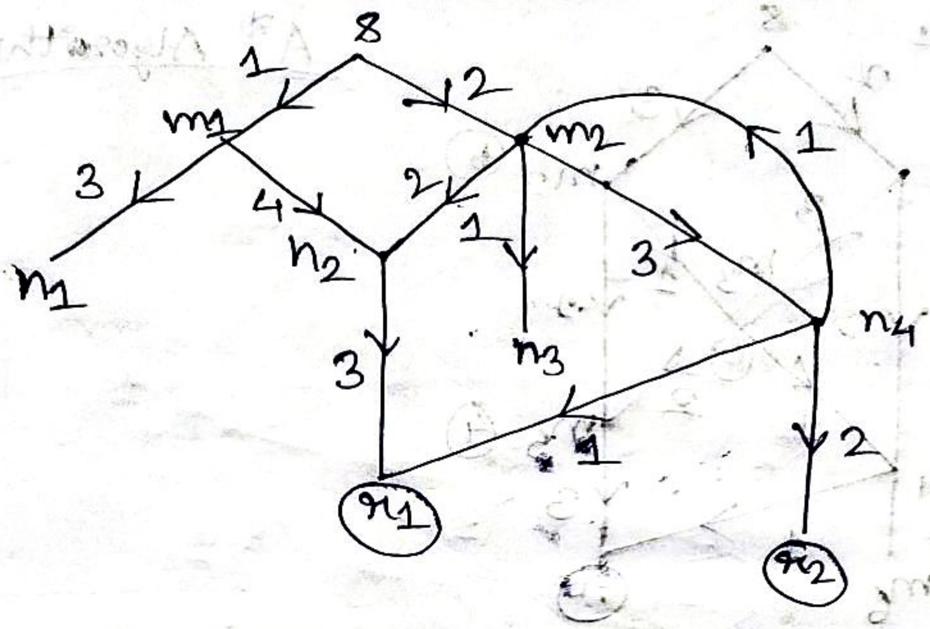
Dijkstra [Uniform Cost]



#	S	m_1	m_2	m_3	m_4	m_5	m_6	r
1	0							
2	$\boxed{0}$	2	3					
3		$\boxed{2}$	3	5	6			
4			$\boxed{3}$	4	6	4		
5				$\boxed{4}$	6	4		
6					6	$\boxed{4}$		7
7					$\boxed{6}$			$\boxed{7}$

Cost = 7

$S \rightarrow m_2 \rightarrow m_5 \rightarrow r$



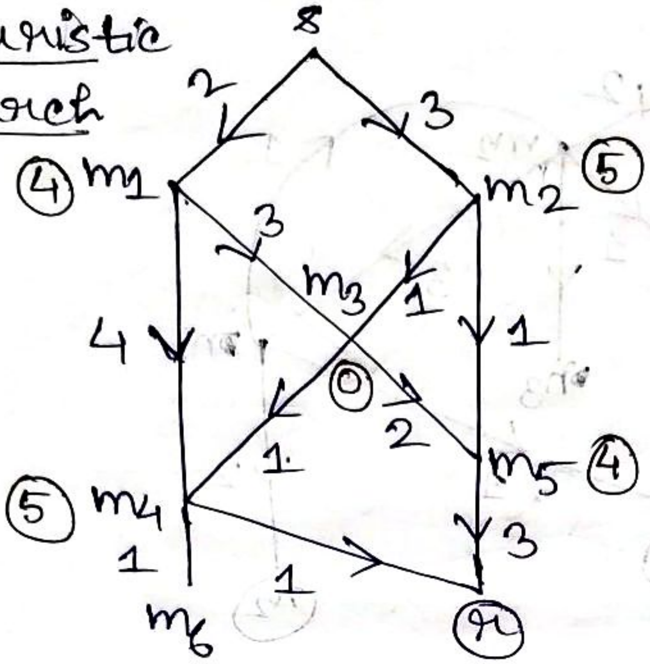
#	s	m_1	m_2	n_1	n_2	n_3	n_4	r_1	r_2
1	0								
2	$\boxed{0}$	1	2						
3		$\boxed{1}$	2	4	5				
4			$\boxed{2}$	4	~~5~~4	3	5		
5			—	4	4	$\boxed{3}$	5		
6					$\boxed{4}$	—	5	7	
7							$\boxed{5}$	6	7
8								$\boxed{6}$	

Cost = 6

$s \rightarrow m_2 \rightarrow n_4 \rightarrow r_1$

But diksktra algorithm does not always ensure the optimal solution.

Heuristic Search



A^* Algorithm

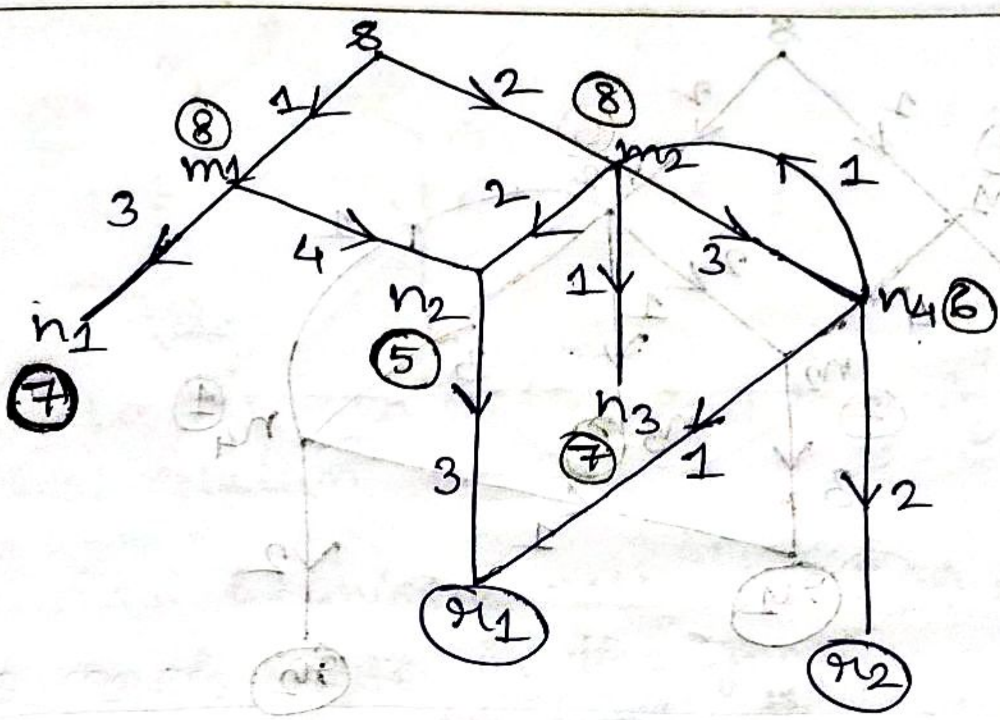
A non negative integer

#	s	m_1	m_2	m_3	m_4	m_5	m_6	r
1	0+0							
2	$\boxed{0+0}$	2+4	3+5					
3		$\boxed{2+4}$	3+5	5+0	6+5			
4			3+5	$\boxed{5+0}$	6+5	7+4		
5			$\boxed{3+5}$	4+0	6+5	4+4		
6				$\boxed{4+0}$	5+5	4+4		
7					5+5	$\boxed{4+4}$		
8								7+0
9								$\boxed{7+0}$

Cost = 7

Total Cost of a node = Actual Cost + Heuristic Cost

$s \rightarrow m_2 \rightarrow m_5 \rightarrow r$

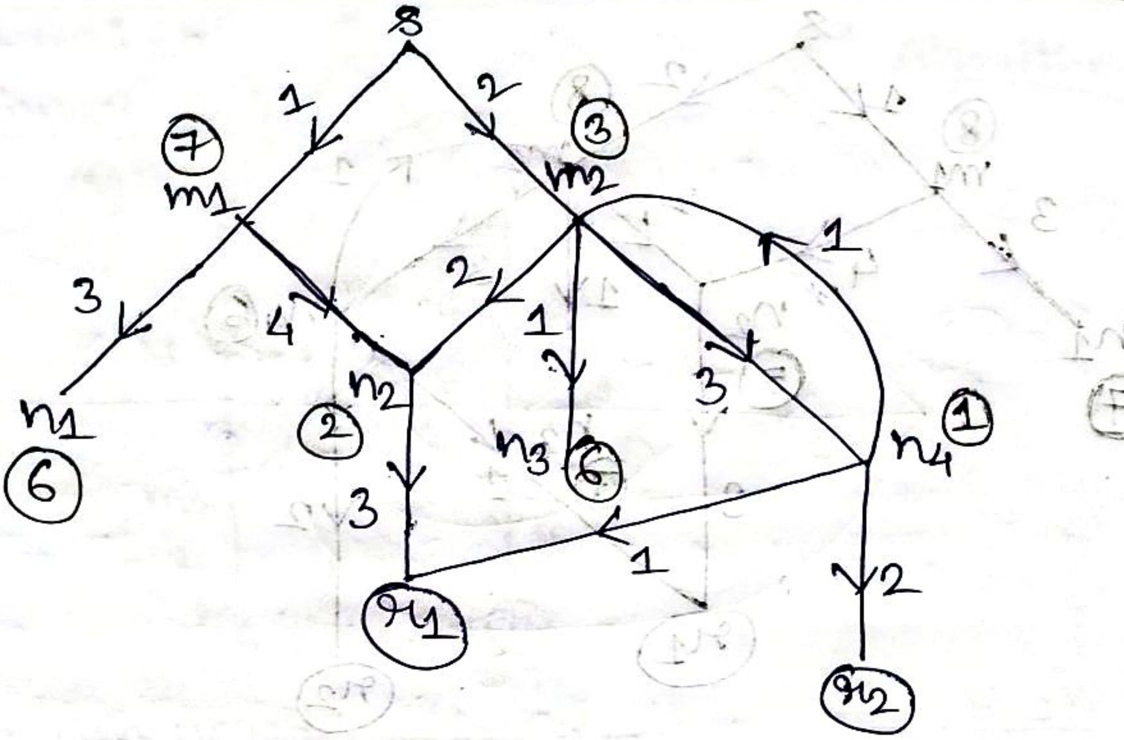


#	s	m1	m2	n1	n2	n3	n4	r1	r2
1	0+0								
2	$\boxed{0+0}$	1+8	2+8						
3		$\boxed{1+8}$	2+8	4+7	5+5				
4			$\boxed{2+8}$	4+7	4+5	3+7	5+6		
5				4+7	$\boxed{4+5}$	3+7	5+6	7+0	
6								$\boxed{7+0}$	

Cost = 7

$s \rightarrow m_2 \rightarrow n_2 \rightarrow r_1$

Optimal Cost is 6



#	s	m1	m2	n1	n2	n3	n4	r1	r2
1	0+0								
2	$\boxed{0+0}$	1+7	2+3						
3		1+7	$\boxed{2+3}$		4+2	3+6	5+1		
4		1+7			$\boxed{4+2}$	3+6	5+1	7+0	
5		1+7				3+6	$\boxed{5+1}$	6+0	7+0
6		1+7				3+6		$\boxed{6+0}$	

Cost = 6

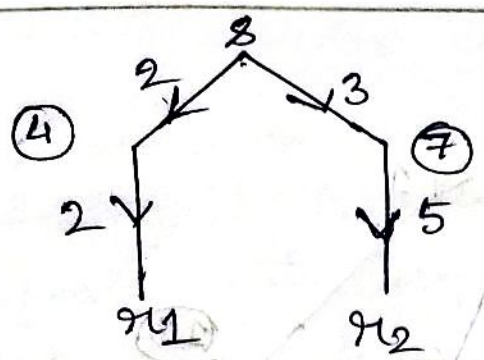
[Which is optimal]

$$0 \le h(n) \le h^*(n)$$

Always gives optimal solution

$h^*(n) \rightarrow$ cost of goal node to that node

$h(n) \rightarrow$ current heuristic value of the node



Heuristic is over estimated till it gives the optimal solution.

- **Properties:**

1) A positive heuristic estimate $h(n)$ is associated with each node in the graph.
2) $f(n) = g(n) + h(n)$ where $g(n)$ is current cost and $h(n)$ is heuristic cost.
3) We define $h^*(n)$ is the actual cost from node n to goal node.
4) $h(n) = 0$ for goal node and source node
   $h(n) = \infty$ for leaf node where goal node does not exist.
5) We define $f^*(n) = g^*(n) + h^*(n)$

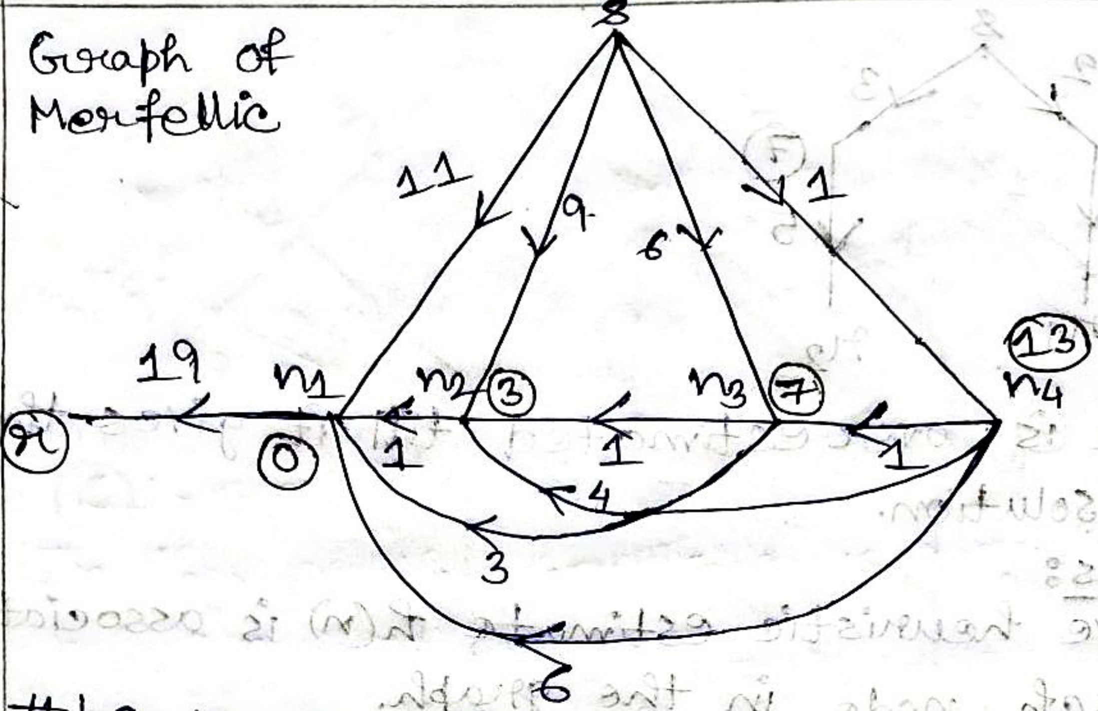
- **Remarks:**

1) A heuristic is called admissible if $0 \leq h(n) \leq h^*(n)$ for all n belongs to G

$$0 \leq h(n) \leq h^*(n) \ \forall \ n \in G$$

2) Algorithm A^* gives optimal solution if heuristic is admissible
3) If the heuristics are inadmissible then we have no idea about the quality of the solution.

Graph of Merfellic



#	s	n_1	n_2	n_3	n_4	r
1	0+0					
2	$\boxed{0+0}$	11+0	9+3	6+7	1+13	
3		$\boxed{11+0}$	9+3	6+7	1+13	30+0
4		10+0	$\boxed{9+3}$	6+7	1+13	30+0
5		$\boxed{10+0}$		6+7	1+13	29+0
6		9+0	7+3	$\boxed{6+7}$	1+13	29+0
7		$\boxed{9+0}$	7+3		1+13	28+0
8		8+0	$\boxed{7+3}$		1+13	28+0
9		$\boxed{8+0}$			1+13	27+0
10		7+0	5+3	2+7	$\boxed{1+13}$	27+0
11		$\boxed{7+0}$	5+3	2+7		26+0
12		6+0	$\boxed{5+3}$	2+7		26+0
13		$\boxed{6+0}$		2+7		25+0
14		5+0	3+3	$\boxed{2+7}$		25+0
15		$\boxed{5+0}$	3+3			24+0
16		4+0	$\boxed{3+3}$			24+0
17		$\boxed{4+0}$				23+0
18						$\boxed{23+0}$

Cost = 23 which is optimal

$s \rightarrow n_4 \rightarrow n_3 \rightarrow n_2 \rightarrow n_1 \rightarrow r$

This is the optimal path

22/8/24

Limitation of A*

1> In worst case the performance is very poor.

2> Repeated node expansion/ A node might be repeated several times.

3> If heuristics are not admissible then we did not reach the goal node/ Optimal solution.

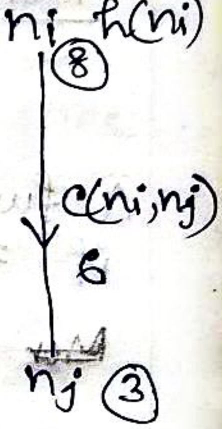
Advantages

Optimal Solution If you choose the right heuristic

Consistency Condition

If $h(n_i) - h(n_j) \le c(n_i, n_j)$ then it is consistent

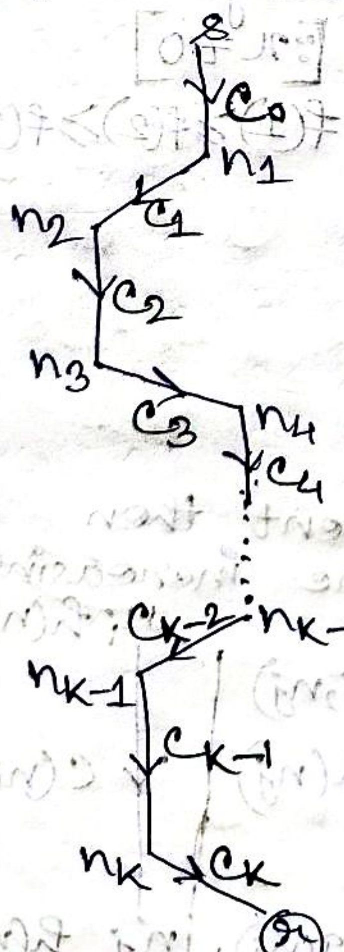
$c(n_i, n_j)$ is the cost of the path n_i to n_j



$8 - 3 \le 6$

Theorem:

If the heuristic is consistent then heuristics are admissible but converse is not true.



Proof: since heuristics are consistent then we can write

$h(n_K) - h(r) \le C_K$

$h(n_{K-1}) - h(n_K) \le C_{K-1}$

$h(n_{K-2}) - h(n_{K-1}) \le C_{K-2}$

— — — — — — — — —

$h(n_2) - h(n_3) \le C_2$

$h(n_1) - h(n_2) \le C_1$

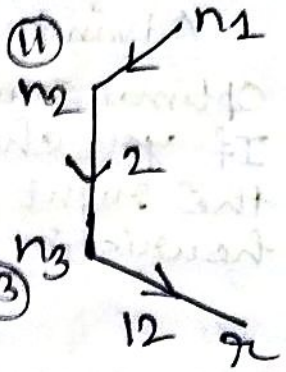
$h(n_1) - h(8) \le C_0$ ×

Adding $h(n_1) - h(r) \le \sum_{j=1}^{K} C_j = h^*(n)$

$\therefore h(n_1) \le \sum_{j=1}^{K} C_j \, (h^*(n))$ $[\because h(r) = 0]$

$0 \le h(n) \le h^*(n)$

Hence heuristics are admissible.



$11 - 3 = 8 \nleq 2$

In this graph heuristic is admissible but not consistent.

Monotone Heuristic

$$f(x) \leq f(x+1)$$

A function or heuristic is called monotone if $f(x) \leq f(x+1)$

Example: Monotone increasing $f(x) = x^2$

$f(1) = 1$
$f(2) = 4$
$f(3) = 9$
$f(1) \leq f(2) \leq f(3)$

Monotone decresing

$f(x) = \frac{1}{x}$ $[\because x \neq 0]$

$f(1) = 1$ $f(1) \geq f(2) \geq f(3) \geq f(4)$

$f(2) = \frac{1}{2}$

$f(3) = \frac{1}{3}$

$f(4) = \frac{1}{4}$

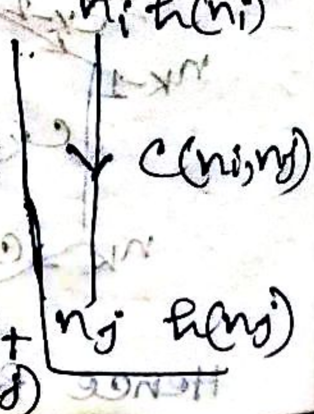
Theorem :

If the heuristic is consistent then heuristic must be monotone increasing.

Prove that $h(n_i) \leq h(n_j)$

clearly $h(n_i) - h(n_j) \leq c(n_i, n_j)$

$h(n_i) - c(n_i, n_j) \leq h(n_j)$



Adding $g(n_j)$ both side

$g(n_j) + h(n_i) - c(n_i, n_j) \leq g(n_j) + h(n_j)$

$$h(n_i) + g(n_j) - c(n_i, n_j) \le f(n_j)$$

$$h(n_i) + g(n_i) \le f(n_j)$$

$$f(n_i) \le f(n_j) \quad [\because g(n_i) = g(n_j) - c(n_i, n_j)]$$

8- puzzle Problem

Initial State

2	8	3
1	6	4
	7	5

Goal state

1	2	3
8		4
7	6	5

$h(n)$

Misplace tiles :-
1, 2, 8, 7, 6

(x coord + y - coord)

$h(n) = 5 = \#$ of misplace tiles

Manhatan dist :-

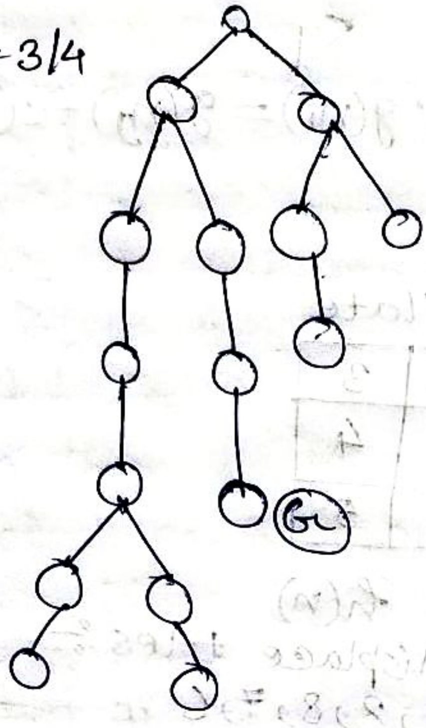
$$(0+1) + (1+0) + (0+0) + (0+0) + (0+0) + (0+1) + (1+0) + (1+1) = 6$$

LEFT
RIGHT
UP
DOWN

29/8/24

Drawback of DFS

L=3/4



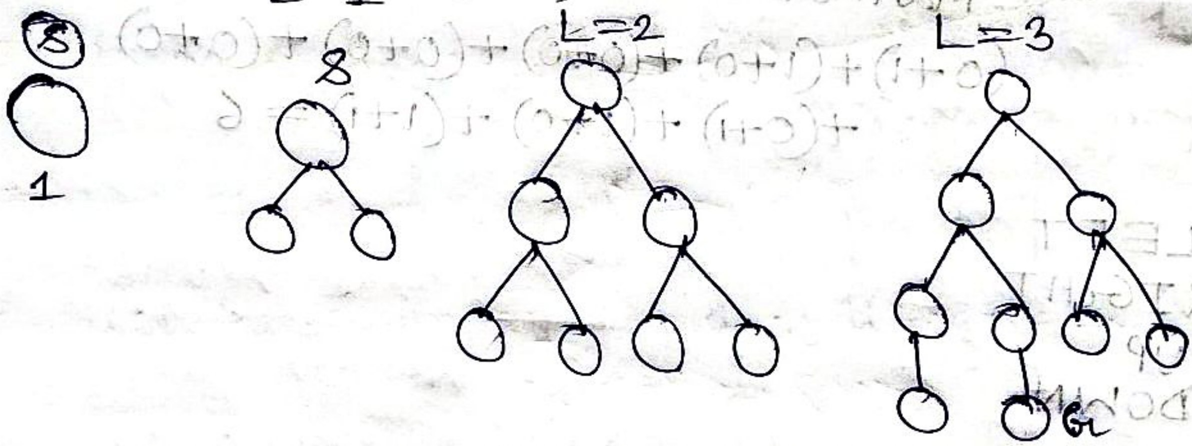
Depth Limited Search (DLS)

Iterative Deeping Search (IDS)

IDS = DLS + BFS

Iterative Deeping Search (IDS)

L=0 L=1 L=2 L=3



Constraints Satisfaction Problem (CSP)

Initial State → Goal State

Given : 1. Variables x_i $(i = 1(1)n)$

2. for each variable x_i, a domain set dx_i $(i = 1(1)n)$ for which x_i can assume a value

3. A set of constraints Example Graph colouring Problem, Line drawing Problem (Scene Analysis), N Queen Problem

	Q		
			Q
Q			
		Q	

		Q	
Q			
			Q
	Q		

Q. A farmer has a wolf, goat, and a cabbage with him. ~~think~~ They are on left bank of the river. He has a boat to ferry them on the other side of the river. He can ferry atmost 1 at a time. But the problem is that he can not leave the wolf and goat together or cabbage with goat. He has to transport all items from left bank to right bank of the river.

→ wgc(Boat Position, left bank, right bank)

wgc(l, [wgc], [])

↓ g

wgc(r, [wc], [g])

↓ alone

wgc(l, [wc], [g])

c ↙ ↘ w

wgc(r, [w], [gc])   wgc(r, [c], [wg])

↓ g   ↓ g

wgc(l, [wg], c)   wgc(l, [cg], [w])

↓ w   ↓ c

wgc(r, [g], [wc])   wgc(r, [g], [wc])

↓

wgc(l, [g], [wc])

↓ g

wgc(r, [], [wgc])

10/9/24

Water Jug Problem

Q. You are given two jugs, a 4L one and 3L one. A pump which has unlimited water, you can used to fill the jug. Water may be poured into the ground (You can waste water). Neither jug has any measuring making on it. How can you get exactly 2 litres of water in 4L of jug.

→ $0 \le x \le 4$
$0 \le y \le 3$

Tuples
(x, y)

Initial State $(0,0)$
Goal State $(2,y) : 0 \le y \le 3$

Operators:

1. Fill 4L of jug: $(x,y) \xrightarrow[x<4]{} (4,y)$
2. Fill 3L of jug: $(x,y) \xrightarrow[y<3]{} (x,3)$
3. Empty 4L of jug: $(x,y) \xrightarrow[x>0]{} (0,y)$
4. Empty 3L of jug: $(x,y) \xrightarrow[y>0]{} (x,0)$
5. Pour water from 3L jug to 4L jug: $(x,y) \rightarrow (4, y-(4-x))$
   $0 < x+y \geq 4$ & $y > 0$
6. Pour water from 4L jug to 3L jug: $(x,y) \rightarrow (x-(3-y), 3)$
   $0 < x+y \geq 3$ & $x > 0$
7. Pour all water from 3L jug into 4L jug : $(x,y) \rightarrow (x+y, 0)$
   $0 \le x+y \le 4$ & $y \geq 0$
8. Pour all water from 4L jug into 3L jug : $(x,y) \rightarrow (0, x+y)$
   $0 \le x+y \le 3$ & $x \geq 0$

#	4L Jug	3L Jug	Rule Applied
0	0	0	—
1	4	0	1. Fill 4
2.	1	3	6. Pour
3	1	0	4.
4.	0	1	8
5	4	1	1
6	2	3	6

Q. 8-Puzzle Problem using Hill Climbing Algorithm — 12/9/24

Predicate Calculus

Knowledge Representation

(1) Predicate

(2) Fuzzy

Proposition :-

$p \equiv$ Grass is green. ✓

$q \equiv$ The sky is blue ✓

$r \equiv 2+2=7$ F ✓

What is your name? ✗

Today is too hot! ✗

Grass is green and cow eat grasses.

Operations ⟶ Connectives

$\sim, \wedge, \vee, \rightarrow, \leftrightarrow$

$\sim p$

$p \wedge q$

Carry umbrella if sky is cloudy.

$p \equiv$ sky is cloudy.

$q \equiv$ Carry umbrella.

$q \rightarrow p$

$p \rightarrow q \Rightarrow \sim p \vee q$

p	q	$p \vee q$	$p \to q$	$\sim p$	$\sim p \vee q$	$p \leftrightarrow q$
F	F	F	T	T	T	T
F	T	T	T	T	T	F
T	F	T	F	F	F	F
T	T	T	T	F	T	T

$$p \leftrightarrow q = (p \to q) \wedge (q \to p)$$
$$(\sim p \vee q) \wedge (\sim q \vee p)$$

Proposition

$p \to q$

$q \to p$ Converse

$\sim q \to \sim p$ Contrapositive

Note: A contrapositive of a proposition is always logical equivalent to proposition.

Types of Proposition

1. Tatulogy 2. Contradiction

3. Contigency

$T(U)$

$F(\emptyset)$

$\vee - \cup \quad \wedge - \cap$

$P \vee T = T$

$P \wedge T = P$

$P \vee F = P$

$P \wedge F = F$

Sentences

Statement

Automic and Complex

$p \vee q \wedge \sim r$

Formula?

Satisfiability - If the value of a formula is true then it is called Satisfiability.

Unsatisfiability/ Contradictory - If the value of a formula is false then it is called Unsatisfiability

Example $p \wedge \sim p = F$

Validity - A proposition formula is called valid or tautology when it is true for all possible interpretetion.

Theorem and Proving

$P_1, P_2, P_3, P_4, \ldots\ldots P_n \vdash C$

Premises Conclusion

1) Forward Chaining
2) Backward Chaining

1) Forward Chaining → When all the premises are true check whether conclusion is true.

2) Backward Chaining → When all the conclusions are false check atleast one of the premises is false.

Example: $p1 \equiv$ The sky is cloudy.

$p2 \equiv$ It will rain.

$p3 \equiv$ If the sky is cloudy then it will rain

$p3 \equiv p1 \rightarrow p2$

$p1, p3 \vdash p2$

$p1$	$\boxed{p2}$	$p3 : p1 \rightarrow p2$
0	0	1
0	1✓	1
1	0	0
1	1✓	1

All premises are true, so the theorem is proved.

p1	p2	p3: p→p2
0	$\boxed{0}$	1
0	1	1
1	$\boxed{0}$	0
1	1	1

All men are mortal. $\forall X$ mortal(X)
Socrates is a man. X= Socrates
Therefore socrates is mortal. mortal(Socrates)

Some men are mortal. $\exists X$ mortal(X)
Sougata is a man. X= Sougata
Bharti is a man X= Bharti
∴ Sougata is mortal. mortal(Sougata)
Bharti is not mortal. ~mortal(Bharti)

object fact { Variable, Constant, Function, Quantifier }

Predicates

$a, x, t(x), t_1(x,y), t_2(a), \ldots\ldots$

1. Constant: a, b, c, John, Sougata, ...
2. Variable: X, Y, Z, ...
3. Functions: f(), g(), h()
4. Operators: $\sim, \vee, \wedge, \rightarrow, \leftrightarrow$
5. Quantifier: $\exists, \forall$
6. Equality: =

eg. Student(Sougata), Married (P, Q)
X, Y: Mother (X, Y)
Child (Y, X)
P = Student(Sougata)

Q. Given a set of people and friendship relation:
friend (N,M) → N is a friend of M

1. Tom is a friend of Bob.
2. Everyone is a friend of everyone.
3. There are two peoples who are friends.
4. There is some person who are friend of everybody.

rates)

1. friend (Tom, Bob)
2. $(\forall X)(\forall Y)$ friend(X, Y)
3. $(\exists X)(\exists Y)$ friend(X, Y)
4. $(\exists X)(\forall Y)$ friend(X, Y)

p: Albert is at home.

19/9/24

q: The door is locked.

if the door is locked then Albert is at home

$q \rightarrow p$

at_home(Albert)

is_locked(door)

is_locked(door) → at_home(Albert)

Convert the following in Predicate

1. Seven is a prime number is odd number. together there is some odd prime numbers.
2. Let the domain be a set of people invited in a meeting. P(x): person x attend the meeting.

   S(x): Person x spoke in the meeting

(a) Jon and Bob attend the meeting.

(b) Everyone was present at the meeting.

(c) Some people spoke at the meeting.

(d) Some people could not come to the meeting.

(e) Some people who came to the meeting but did not spoke.

(f) Everyone who came to the meeting spoke on some topic.

1. $[Prime(7) \wedge odd(7)] \rightarrow [(\exists x) Prime(x) \wedge odd(x)]$

2. (a) $P(John) \wedge P(Bob)$

(b) $(\forall x)\, p(x)$

(c) $(\exists x)\, S(x)$

(d) $(\exists x)\, [\sim p(x)]$

(e) $(\exists x)\, [p(x) \wedge \sim S(x)]$

(f) $(\forall x)\, [p(x) \rightarrow S(x)]$

Sougata is in lab then it must be Software Lab.

H.W

1. Every city has a dog catcher who has been beaten by every dog in the town.

2. Resulation

All men are mortal.
John is a man.
So, $\boxed{\text{John is mortal.}}$ ← To proof

Step 1: Convert all sentences into predicate form.

$(\forall x)\, mortal(x)$ [mortal(x): x is mortal
man(John) man(x): x is man
mortal(John) $(\forall x)\, [man(x) \rightarrow mortal(x)]$]

Step 2: Clause form (All premises should be added with AND clause

$(\forall x)\, mortal(x) \wedge man(John)$, $mortal(John)$

Remove all quantifiers, $\rightarrow$, $\leftrightarrow$

$\sim mortal(x) \vee$ ' $\sim man$ '

$\sim man(x) \vee mortal(x)$, $man(John)$, $mortal(John)$

<u>Step 3</u> : Let, S be the set of WFF except conclusion

$S = \{\sim man(x) \vee mortal(x), man(John)\}$

<u>Step 4</u> :- Consider only conclusion

$L = mortal(John)$

Take its negation

$L' = \sim mortal(John)$

$S' = S \cup L'$

$= \{\sim man(x) \vee mortal(x), man(John), \sim mortal(John)\}$

<u>Step 5</u> :

$\sim man(x) \vee mortal(x)$ $\qquad$ $man(John)$

$\sim mortal(John)$

$\{John/x\}$

$\cancel{\sim man(John)} \vee mortal(John) \wedge \cancel{man(John)}$

$mortal(John)$

$[\phi]$

Conclusion is followed by premises.

1. All lecturers are determine.
2. Anyone who is a determine and intelligent give good service
3. Mary is an intelligent lecturer.

Show that Mary will give good service.

Step 1: $r(x)$: x is lecturer.
$d(x)$: x is determine.
$i(x)$: x is intelligent.
$g(x)$: x gives good service.

$(\forall x)\, r(x) \rightarrow d(x)$
$(\forall x)[d(x) \wedge i(x) \rightarrow g(x)]$
$r(\text{Mary}) \wedge i(\text{Mary})$
$g(\text{Mary})$

Step 2: $\sim r(x) \vee d(x)$
$\sim d(x) \vee \sim i(x) \vee g(x)$
$r(\text{Mary}) \wedge i(\text{Mary})$
$g(\text{Mary})$

Step 3: $S = \{\sim r(x) \vee d(x), \sim d(x) \vee \sim i(x) \vee g(x), r(\text{Mary}) \wedge i(\text{Mary})\}$

Step 4: $L' = \sim g(\text{Mary})$
$S' = \{\sim r(x) \vee d(x), \sim d(x) \vee \sim i(x) \vee g(x), r(\text{Mary}) \wedge i(\text{Mary}), \sim g(\text{Mary})\}$

Step 5:

$\sim r(x) \vee d(x)$ $\quad\quad$ $\sim d(x) \vee \sim i(x) \vee g(x)$

$\sim r(x) \wedge \sim i(x) \vee g(x)$ $\quad\quad$ $r(Mary)$

$\{Mary|x\}$

$\sim i(Mary) \vee g(Mary)$ $\quad\quad$ $i(Mary)$

$\sim g(Mary)$ $\quad\quad$ $g(Mary)$

[ ]

1. Mary, Micky & John are member of Rotary club.
2. Every rotary club member who is not swimmer is a mountain climber.
3. Mountain climber do not like rain.
4. Anyone who does not like water is not swimmer.
5. Micky dislike whatever Mary like and like whatever Mary dislike.
6. Mary like rain & water.

Is there any member who is not a mountain climber but a swimmer?

mem(x): member of rotary club.
mc(x): x is mountain climber
s(x): x is a swimmer.
like(x,y): x likes y.

$mem(Mary), mem(Micky), mem(John)$

$(\forall x)[[(mem(x) \land \sim s(x)] \rightarrow mc(x)]$

$(\forall x)\ [mc(x) \longrightarrow \sim like(x, rain)]$

$(\forall x)[\sim like(x, \overset{water}{\cancel{rain}}) \longrightarrow \sim s(x)]$

$(\forall y)\ [like(Mary, y) \longrightarrow \sim like(Micky, y)] \land$

$(\forall z)\ [\sim like(Mary, z) \rightarrow like(Micky, z)]$

$like(Mary, water), like(Mary, rain)$

To proof $mem(x) \land \sim mc(x) \longrightarrow s(x)$

<u>Step 2</u> : $mem(Mary), mem(Micky), mem(John)$

$\sim mem(x) \vee s(x) \vee mc(x)$

$\sim mc(x) \vee like(x, rain)$

$like(x, water) \vee \sim s(x)$

$\sim like(Mary, y) \vee \sim like(Micky, y) \land$

$like(Mary, z) \vee like(Micky, z)$

$like(Mary, water), like(Mary, rain)$

$\sim mem(x) \vee mc(x) \vee s(x)$