422 Introduction :

AI = ? Artificial → not natural, human made.
Intellegence → $\boxed{\text{learn}}$ from environment, understand that and makes decision.

human being → complex decision making, complex cognitive task perform, cope with complex environment

these decision making skills, complex cognitive ability are part of intelligence.
- → ability to perform complex cognitive tasks
- → using experiences
- → self awareness
- → decision making.

When these intelligence is integrated in a machine or agent (an entity that takes input, processes it, produce outputs) is called Artificial Intelligence.

AI is within an agent. (Physical / Non-Physical) :
- → input
- → process
- → output

Scanned with CamScanner

Pillars of AI (goals of AI)

AI → stores knowledge. (experience)

↳ Reason about " . (knowledge কে কাজে লাগিয়ে further কোনো task accomplish করা)

↳ Behave intelligently.

(complex decision making procedure)

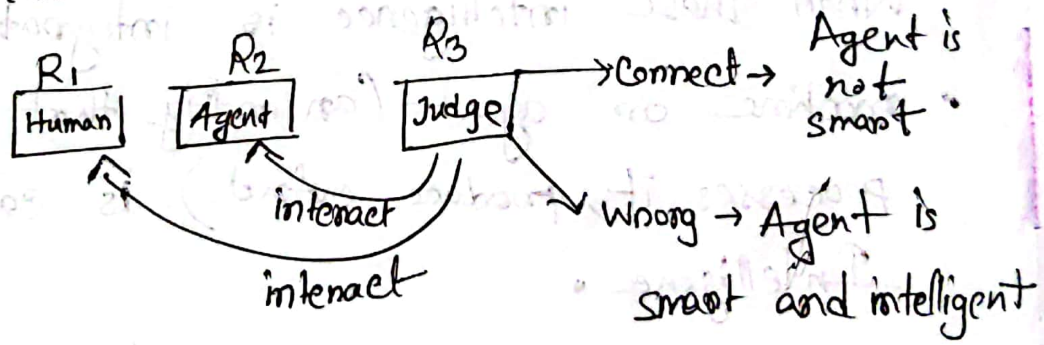
↳ game playing

↳ math solve

↳ ability to interact with people, agents and environment

* 1956 → "AI" word. (Darthmouth University)

* Turing Test :



Late 1970 : First AI winter

Late 1980 : Second " "

After 2000 : Facial recognition, Image Processing, Object ", detection.

(Game ↓ Hard ware advance ment.)

CNN → Alexnet (Deeplearning framework)

↳ Image থেকে object detection.

Scanned with CamScanner

AI advancement → cons → i) job replacement

ii) Rationality → Goal achieve

AI 's

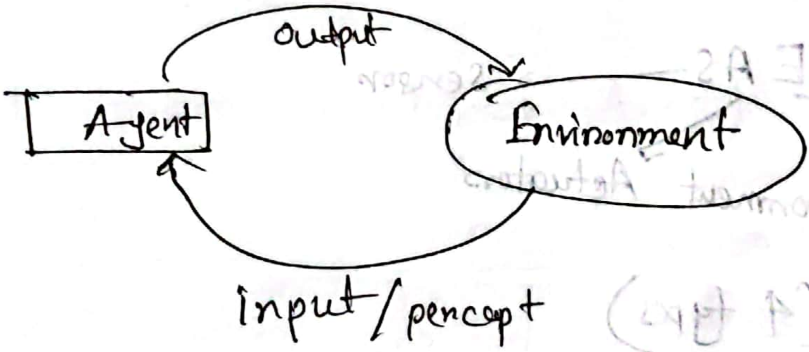
↓

in efficient way (not always correct)

AI Behaviour → Human like behaviour — Think like human — act " "

→ Rational behaviour (achieve the goal in any way) — think rationally — act "

Agent :



Human is a type of Agent, Robot, animal, } has physic entity.

self driving car.

Google Search engine → Non-physical ~~engines~~ agent.

input : Search keywords

output : " results.

Environment : www

Human → Sensor (input)

↳ Eyes, ears, nose, skin

Actuators (output)

Human → Legs, hands

Envioronment -

Performance measurement : matrix for performance evaluation.

↑

PEAS → Sensor.

↙ Environment ↘ Actuators

Agent (4 types)

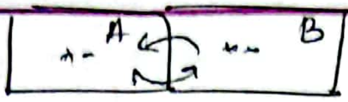
1) Simple Reflex agent.

Instantaneous input ↳ u output. } decision is based on current moment.

doesn't consider prev environment state / future " " .

↳ Table driven agent

lookup table



Percept	Action
A clean	Move to B
B clean	Move to A
Dirt in A	Clean A
Dirt in B	Clean B

Disadvantage:

→ Infinite loop

→ table is insufficient for complex environment

~~too large~~

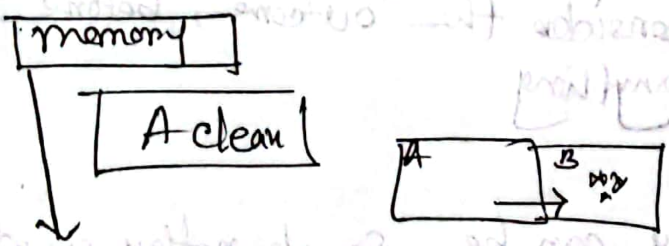
→ table might be too large

→ Doesn't handle environment changes well

Advantage

→ easy to implement

) Agent with memory (Model based Agent)



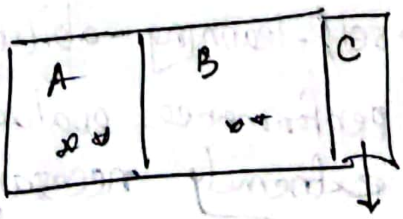
it can also be table driven.

unperceived environment as state in memory, so that extra functionality can be achieved

3) Agent with Goal.

Action: 2 possible.

Goal: Clean all the tiles.



Move to other tile

Clean the tile.

→ number of tiles with dirt → 0

dynamic: new tiles can be added.

Disadvantage:

→ purely goal based agent → rational agent.

↓ achieve goal anyhow

Most advanced type of agent:

④ Utility Based Agent:

↳ Defined Goal

↳ performation evaluation/matrix measurement.

extra → utility fun^c, which defines the measurement of happiness.

~~Goal~~ → consider the outcome, before doing anything.

Agent with memory $\xrightarrow{\text{can be}}$ Simple reflex agent.

Utility based agent → Agent with goal

Bottom to top · agents

4 → 1.

Learning Agent: → self-learning ability.

→ performance evaluation is extremely necessary.

Reinforcement learning → ভুল থেকে ~~[illegible]~~, outcome থেকে learn'

Properties Environment :

⇒ Fully or Partially Observable. → portion of the environment

↓

If the agent's sensors can percieve / observe the full environment. /have access over the environment.

⇒ Single or Multi-Agent Environment :

⇒ Deterministic or stochastic (ludo)
(chess)

Agent taking an action.
↳ certain outcome → uncertain outcome.

(simple reflex agent)
⇒ episodic or sequential. (most of the games) complex environment real world .. .
↳ environment's steps have no connection within each other.
↳ environment ~~or~~ change.
→ series of events, dependent on each other.

⇒ discrete or continuous (Real world, self driving car)
Poker
→ unlimited or undefined number of perception and possible actions.
→ limited. Chess. 10^{120} moves.

turn based
(game)

⇒ Static or Dynamic
(Chess) (Real world)
↓
environment এ
sudden change না আসে

⇒ known or unknown → learning agent
↓
(একটা environment এর
perception, " এ কিসের
action নেওয়া যায়, এটা
সম্পর্কে given agent এর
complete idea থাকে)

Scanned with CamScanner

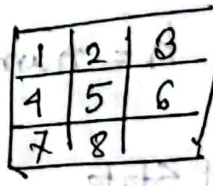
Uninformed Search:

BFS, DFS

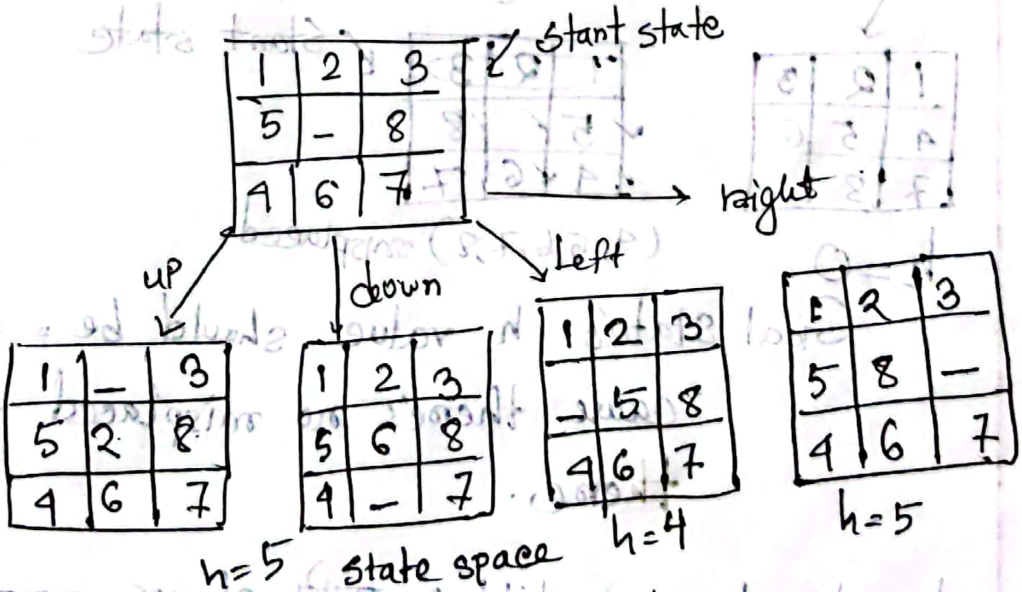
Depth limited Search (DLS)
Iterative deepening Search (IDS)
Uniform Cost Search
Bi-directional Search

Informed Search
→ Greedy Best first Search
→ A*

State Space : 8 puzzle



Goal State



State space

In uninformed search, tree → nodes, edges doesn't know how to expand (whether to go top, down.. ..) so that we can reach the goal.

→ TC, SC বেশী
example: BFS → b^d
DFS → b^m
→ exponential TC.

→ no extra info, or rule set to follow
→ কোন branch এ progress হলে goal পাওয়ার possibility বেশি, blindly search

So, to overcome,

→ we'll provide the searching algo with more info.

→ To find the goal state, it'll go forward, based upon the following criteria,

Approximated
* Distance from the goal state
  ↳ heuristic(h)

We'll compare with the goal node, and calculate h here, for example,

h = number of misplaced tiles.

Goal State

$h = 1+1+1+1+1$
$= 5$

Start state

1	2	3
4	5	6
7	8	

1	2	3
5		8
4	6	7

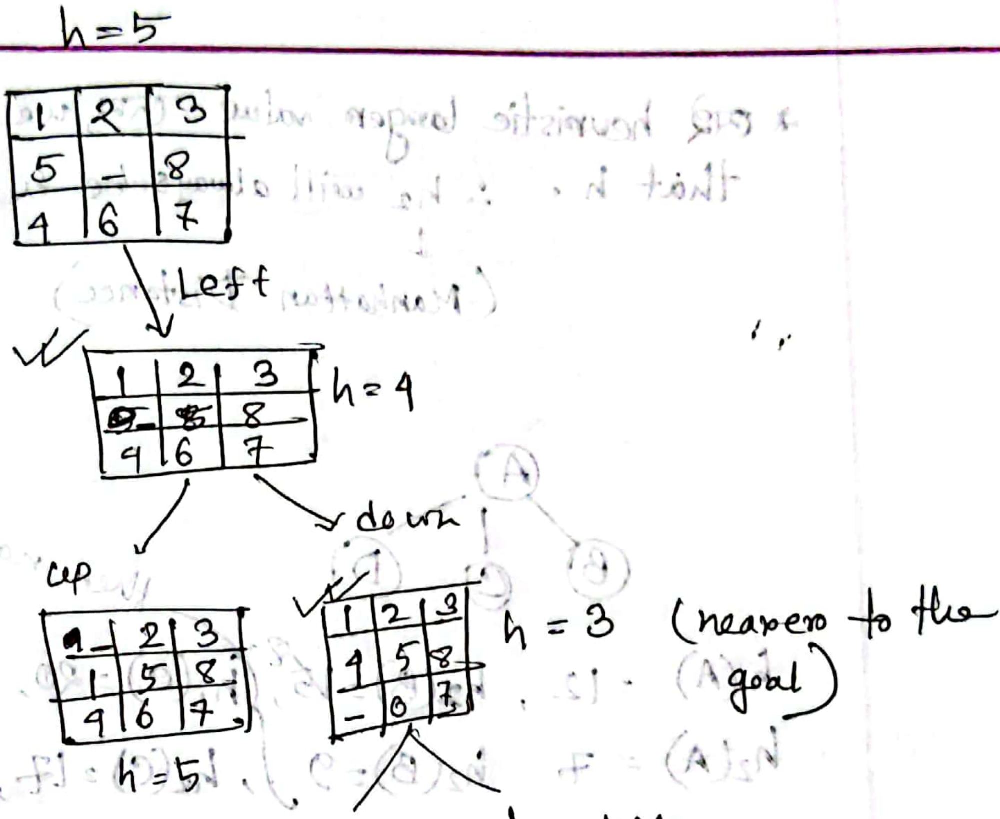
$h = 0$

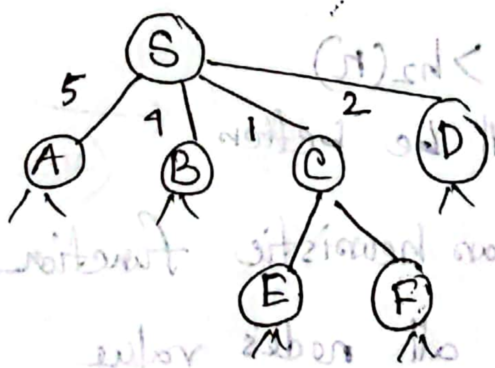
(4, 5, 6, 7, 8) misplaced

Goal state's h value should be = 0 cause there's no misplaced tile there.

h value low যেই child এ, সেটা choose করবো,

∴ $h=4$ will be expanded.





h - table

node	h-value
S	5
A	6
B	5
C	4
E	5
F	3
⋮	
G	0

8 puzzle

↪ $h_1 \Rightarrow$ No of misplaced tiles

↪ $h_2 \Rightarrow$ Manhattan distance

1	2	3
4	5	6
7	8	

1	3	2
4	7	8
6	5	

$h_1 = 6$

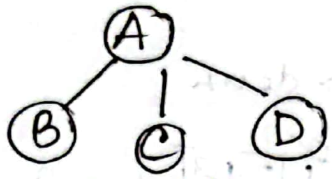
$$h_2 = \underset{3}{1} + \underset{2}{1} + \underset{7}{2} + \underset{8}{2} + \underset{6}{3} + \underset{5}{1}$$

$= 10$

(amount of misplacement) ←

vertical and horizontal, not diagonal

* সেই heuristic larger value দেয়, we choose that h. ∴ h_2 will always be better.
↓
(Manhattan Distance)



then no dominance

$h_1(A) = 12$, $h_1(B) = \overset{8}{15}$, $h_1(C) = 20$, $h_1(D) = 9$

$h_2(A) = 7$, $h_2(B) = 9$, $h_2(C) = 17$, $h_2(D) = 8$

here; $h_1(n) > h_2(n)$

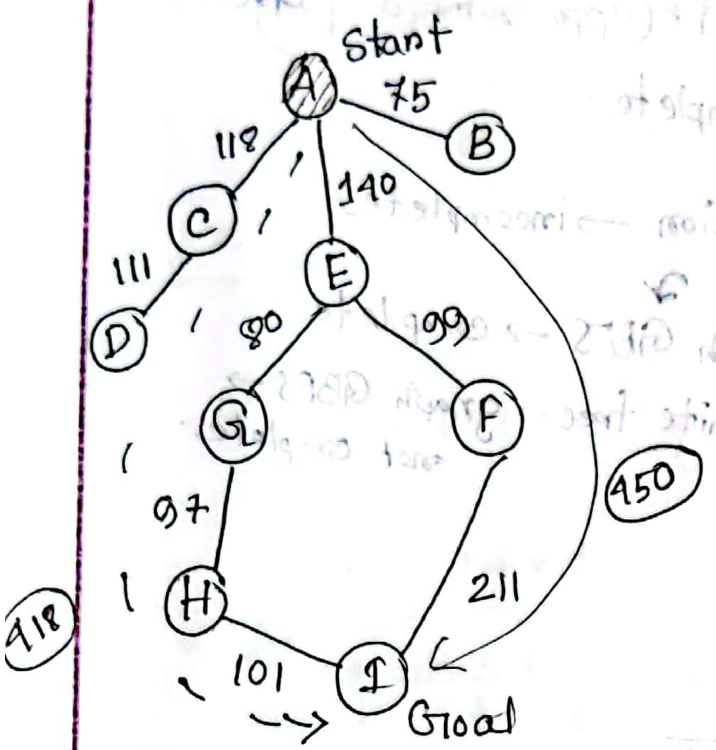
∴ h_1 will be better.

As, for ~~this~~ two particular heuristic function, when a ~~node~~ tree's all node's value is larger than another, or equal it's called dominance.

∴ here, h_1 dominates h_2.

$h_1 \geq h_2$

Greedy Best First Search (not optimal)



h- table

state	h(n)
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

$f(n) = h(n) =$ straight-line distance heuristic.

$A \to E \to G \to H \to I$

$\Rightarrow 140 + 80 + 97 + 101$

$= 418$

Fringe (Priority Queue) → $\boxed{A^{366}}$

A^{366} childs of A → $\boxed{C^{329} \quad E^{253} \quad B^{374}}$

(lowest heur) E^{253} $\boxed{C^{329} \quad B^{374} \quad G^{193} \quad F^{178}}$

F^{178} $\boxed{C^{329} \quad B^{374} \quad G^{193} \quad I^{0}}$

I^{0} $\boxed{C^{329} \quad B^{374} \quad G^{193}}$

Parent-child relation

$A \longrightarrow E \longrightarrow F \longrightarrow I$

$140 + 99 + 211$

$= 450 > 418$

$\therefore$ GBF search cannot give optimal path.

Scanned with CamScanner

GBFS → not optimal. (approximated path)

Complete? → not complete.

tree version → incomplete

graph " " → complete

for, finite tree → graph GBFS → complete

tree GBFS : for infinite tree → graph GBFS → not complete.

$\boxed{A^{366}}$

A^{366} $\boxed{B^{374} \; C^{250} \; E^{253}}$

C^{250} $\boxed{B^{374} \; E^{253} \; A^{366} \; D^{244}}$

D^{244} $\boxed{B^{374} \; E^{253} \; A^{366} \; C^{250}}$

C^{250} $\boxed{B \; E \; A \; D^{244}}$

< C D বার বার push and pop cause bidirectional and falls into a infinite loop.

But graph search prevents loop.

$\boxed{A^{366}}$

A^{366} $\boxed{B^{374} \; C^{250}}$ E^{253}

C^{250} $\boxed{B^{374} \; E^{253} \; D^{244}}$ (won't push A, cause A already expanded)

D^{244} $\boxed{B^{374} \; E^{253}}$

Time & Space Complexity → $O(b^m)$ = similiar to DFS

b → branching factor

m → max depth.

A* search : (GBFS ~~or~~ extension)

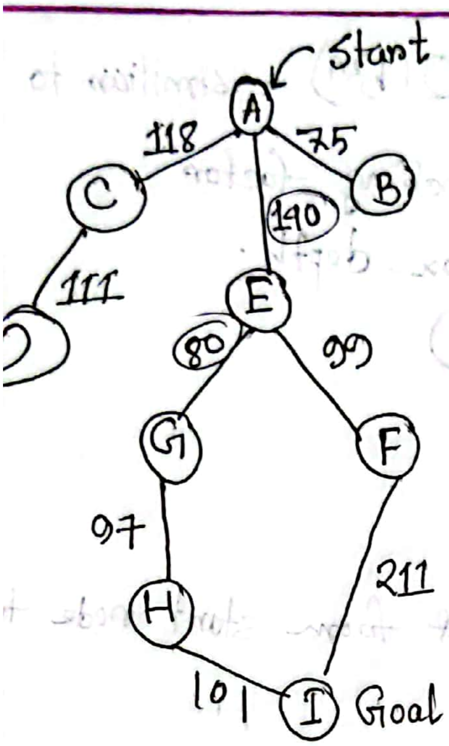
$$A^* \rightarrow f(n) = \underbrace{h(n)}_{\text{heuristic}} + \underbrace{g(n)}_{\text{path cost from start node to node } n}$$

∴ * A* → GBFS + Uniformed Cost Search

if h(n) → becomes 0, A* works like Uniform cost search.

if g(n) → " 0, A* " " GBFS.

A* search ($f(n) = h(n) + g(n)$)



State	h(n)
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0

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

$= 366 + 0$

$= 366$

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

$= 329 + 118$

$= 447$

Fringe

$\boxed{A^{366+0=366}}$

A^{366} $\boxed{C^{447} \quad E^{253+140=393} \quad B^{374+75=449}}$

E^{393} $\boxed{C^{447} \quad B^{449} \quad G^{193+140+80=413} \quad F^{417}}$

G^{413} $\boxed{C^{447} \quad B^{449} \quad F^{417} \quad H^{98+317=415}}$

H^{415} $\boxed{C^{447} \quad B^{449} \quad F^{417} \quad I^{0+418}}$

F^{417} $\boxed{C^{447} \quad B^{449} \quad I^{418} \quad I^{0+450=450}}$

I^{418} $\boxed{C^{447} \quad B^{449} \quad I^{450}}$

$A \longrightarrow E \longrightarrow G \longrightarrow H \longrightarrow I$

Cost = 418 (optimal)

A* search's optimality depends on heuristic's $\boxed{\text{admissibility}}$

① Admissibility : h হচ্ছে কোনো একটি node থেকে goal এ যাওয়ার approx cost.

for every node, from that node to goal, approx cost (h) should be lower than ঐ node থেকে goal এ যাওয়ার actual cost.

সবগুলো node এর জন্য এটা হলে, the graph or tree's heuristic will be admissible.

H

↳ h cost ⇒ 98

↳ actual cost ⇒ 101

∴ $98 \le 101$

F

↳ h cost ⇒ 178

↳ actual cost ⇒ 211

∴ $178 \le 211$

E

↳ h cost ⇒ 253

↳ actual cost ⇒ $\boxed{278}$ not 310

will consider the lower one

$253 \le 278$

∴ the graph is $\boxed{\text{admissible}}$

even if the condition brakes for a single node, the graph is not admissible.

if $h(H) \Rightarrow \cancel{98} \rightarrow 138$

then, for H

↳ h cost ⇒ 138

↳ actual cost ⇒ 101

∴ 138 ≥ 101

∴ h is overestimated.

* admissible না হলে, A* returns suboptimal root.

Fringe

A^{366}

A^{366}	B^{449} C^{447} E^{393}
E^{393}	B^{449} C^{447} G^{413} F^{417}
G^{413}	B^{449} C^{447} F^{417} H^{455}
F^{417}	B^{449} C^{447} H^{455} I^{450}
C^{447}	B^{449} H^{455} I^{450} D^{473}
B^{449}	H^{455} I^{450} D^{473} (no expand)
I^{450}	H^{455} D^{473} (goal expanded) so terminate

we found the sub-optimal cost here.

$$A \rightarrow E \rightarrow F \rightarrow I$$

$$\text{cost} = 450$$

$\therefore$ A^* search's admissibility depends on heuristic.

$\therefore$ Ques: Is A^* search optimal?

Ans: If h is admissible $\rightarrow$ optimal

if h not $\rightarrow$ not optimal (tendency of returning a sub-optimal path)

② & ③ Time and Space complexity:

$$O(b^{\epsilon d})$$

visually looks like uninformed search's complexity (exponential)

but in actual implementation time and memory can get significantly reduced depending on the quality of heuristic value. (ϵ)

④ heuristic consistency:

$$\overset{h(A)}{A} \xrightarrow{C(A,B)} \overset{h(B)}{B} \xrightarrow{C(B,G)} \overset{h(G)}{G}$$

to be consistent, $h(n) \leq h(n') + c(n, n')$ for each node except for goal node

n' — child

So here, $h(A) \leq h(B) + C(A,B)$

$h(B) \leq h(G) + C(B,G)$

* heuristic can be admissible, but not consistent.

$h=4$ $h=1$ $h=0$

$$(A) \xrightarrow{1} (B) \xrightarrow{3} (G)$$

$h(A) \le h(B) + c(A,B)$

$4 > 1 + 1$

$\therefore$ not consistent. (TC and SC will increase)

but, for $\underline{A}$

↳ h cost $= 4$

↳ true cost $= 3+1 = 4$

$4 \le 4$

$\underline{B}$

↳ h cost $= 1$

↳ true cost $= 3$

$1 \le 3$

$\therefore$ admissible.

⑤ Completeness:

↳ Graph Search → h ~~consistent~~ admissible → complete

↳ Tree Search → h both admissible + consistent → complete.

Local Search

In prev searches, → Backtracking was needed
→ ~~Add~~ Path mattered (infinite search space)
→ Perfect goal

Local Search — advantage
- Not concerned with path.
- Solution itself matters.
- Not concerned with perfect sol^n.

→ as no backtracking, Less memory (typical Local search → no memory)
no node store,

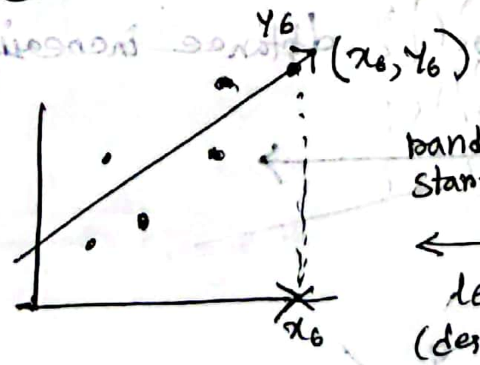
→ Less time

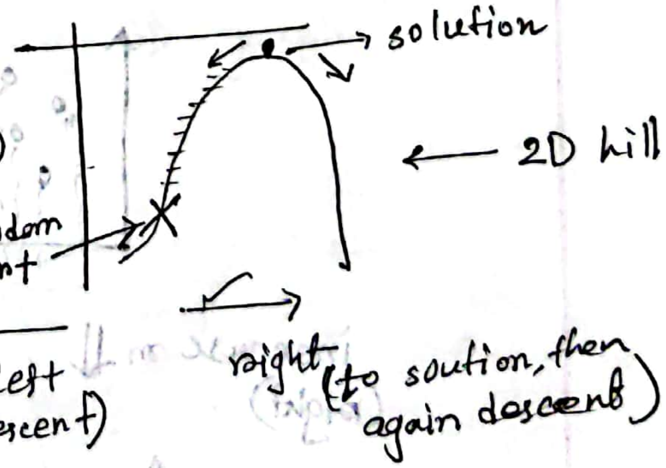
→ Applicable for large size graph.

→ Hill Climb Search :

x_1	y_1
x_2	y_2
x_3	y_3
x_4	y_4
x_5	y_5
x_6	?

Linear Regression





$y = mx + b$

↑ slope ↑ intercept

$y = mx$ (centered line) → $\boxed{\text{search}}$

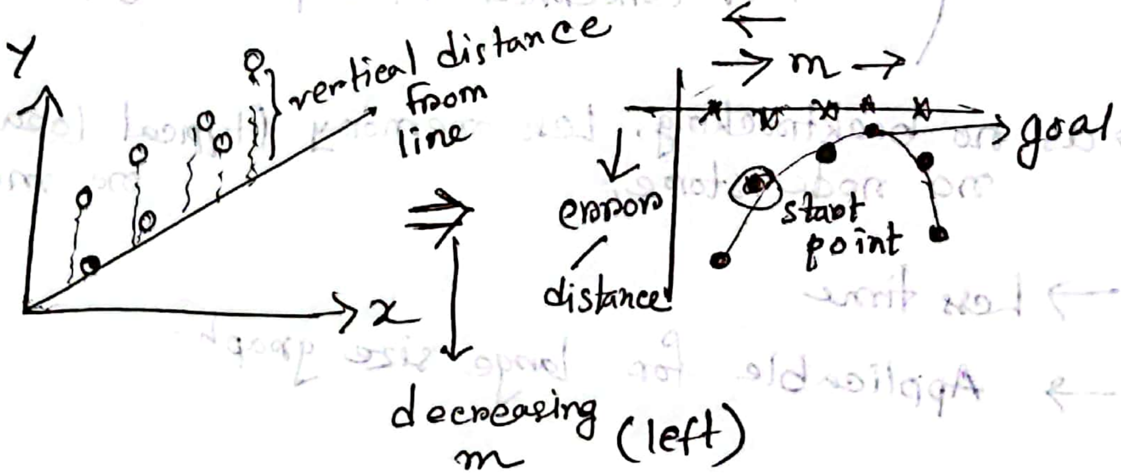
known, unknown, known

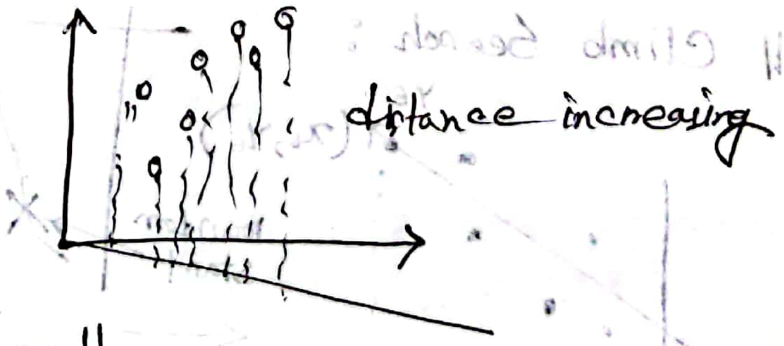
in perspective of state space tree/graph,
this search space is infinite,
because,

m is a variable
and $-\infty \le m \le \infty$ (values)
$\downarrow$
∴ infinite search space.

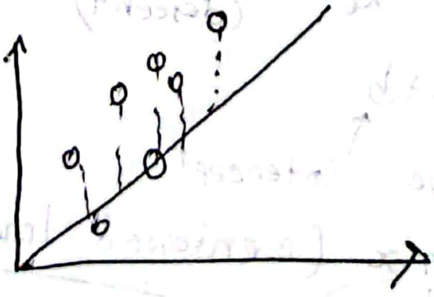
(Local Search) but here, only solution matters.

⇒ initialize m randomly.



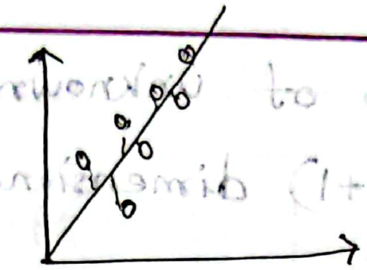


increase m ⇓
(right)

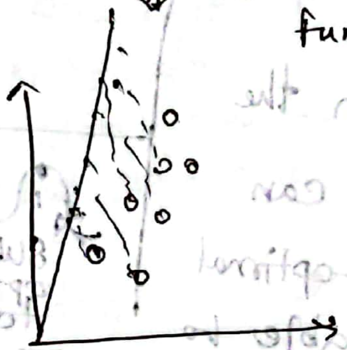


GBFS → Fringe → Back-tracking
HC → No " → No "

⇒ increase m



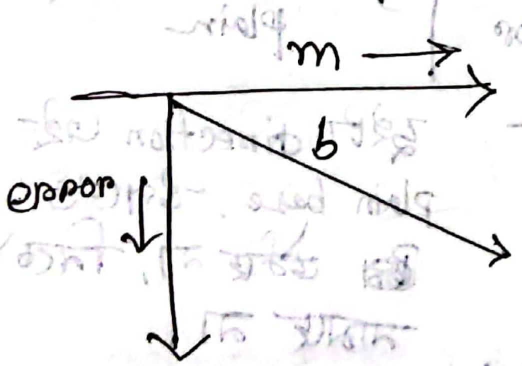
⇓ increment m further



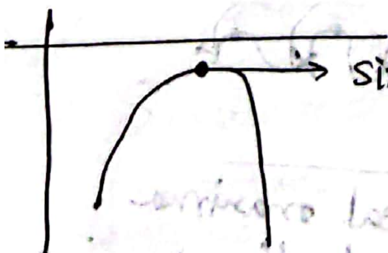
Hill climbing example : regression analysis in iterative way.

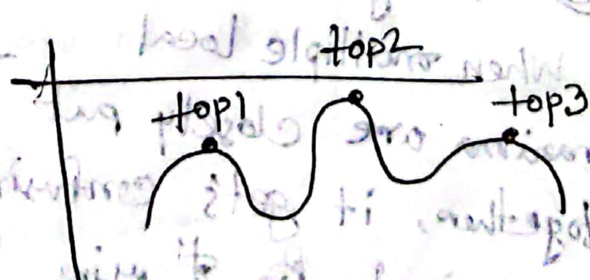
$y = mx + b$

↓ unknown , ↘ unknown



3D bowl shape from this 3D graph.

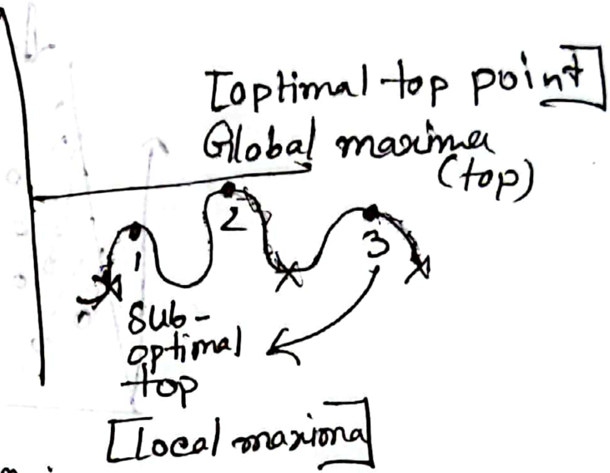




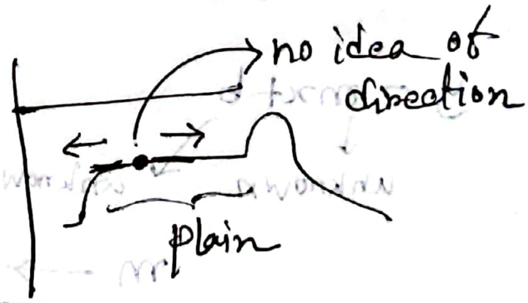
n-টা number of unknown থাকলে, it'll be (n+1) dimensional.

Disadvantages:

① Depending on the start point, it can return the sub-optimal top. won't be able to find the global maxima.
↳ Local maxima.

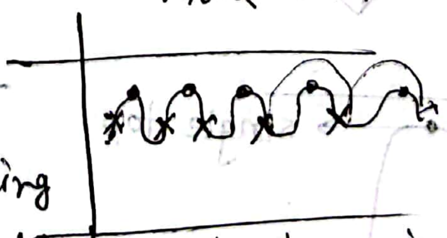


② Plateaus →
plain land এ আটকে যাওয়া, finding no right or left path for going to the goal.



দুইটা direction এই plain base, উঠতেও উঠছে না, নিচেও নামছে না।

③ Ridges.
When multiple local maxima are closely put together, it get's confusing to navigate using iterative approach.



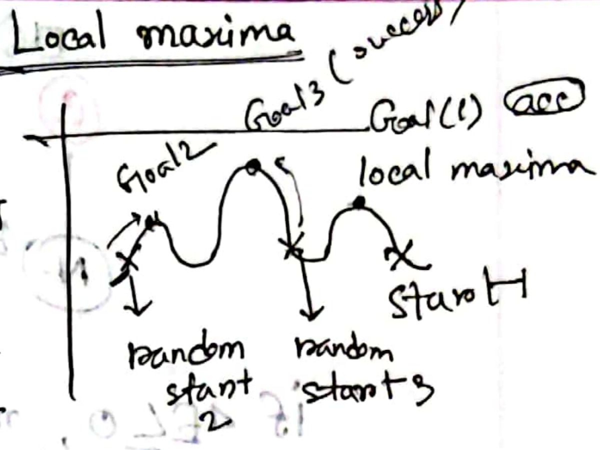
multiple local maxima closely. constantly jumping but no improvement.

Ridges occurs, when there are close proximity of local maxima.

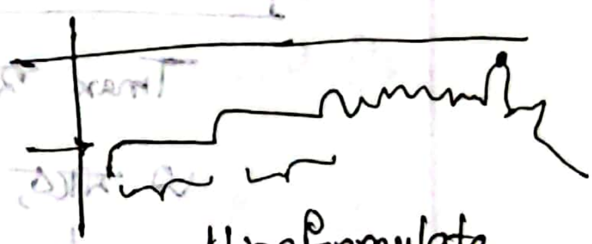
Solution :

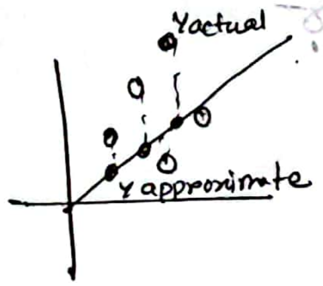
↳ Random restart.

eg. (goal এর % কাছাকাছি পৌছলে আর random restart নেবো না, set max itr value (at most 50k random restart) → যদি এই itr এর মধ্যে global maxima না পাওয়া যায়, then terminate.

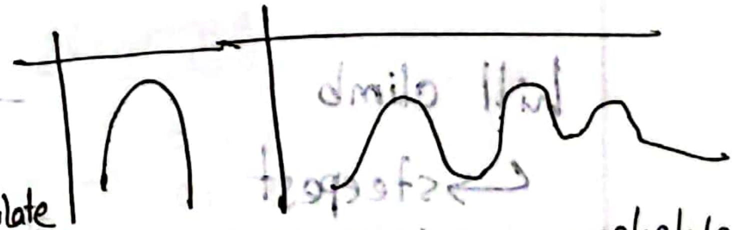


↳ Problem reformulation



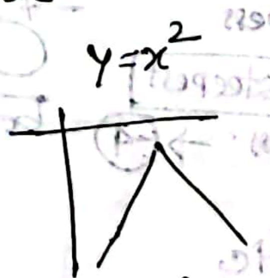


$\Sigma (y_{acc} - y_{app})^2 \Rightarrow$ formulate hill



mod

$\Sigma |y_{acc} - y_{app}| \Rightarrow$



∴ reformulating the problem, changes shape of the hill.

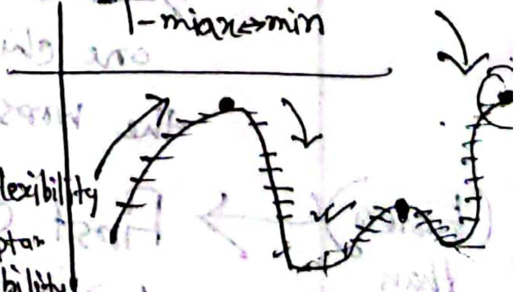
↳ Simulated Annealing :

temperature val

↳ high to low

temp → higher → more flexibility → error acceptance

→ lower → less flexibility



⇒ hill climbing এ descent করতো না

in here, we add tolerance based on temperature

so tolerance high থেকে low তে গেলে, it won't further descent.

fitness value

E(C) → energy val / goodness value

parent state / initial

(C)

E(N) > E(C) হলে (C) → (N)

(N) child node

if $\Delta E \le 0$, means $E(N) \le E(C)$

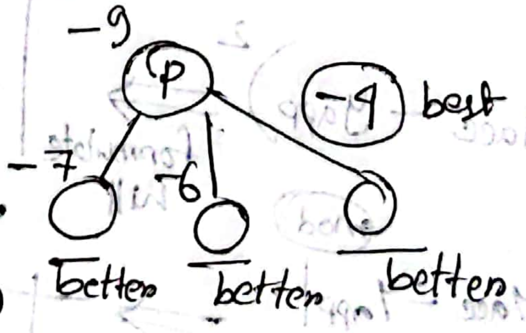
$\boxed{\text{else if } e^{\Delta E/T} > (\text{rand}(0,1))}$

Tmax হওয়া পর্যন্তই else if এ যাবে, Tmin হওয়ার পর accept করবে না।

hill climb

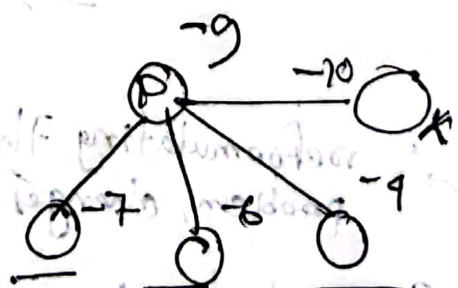
↳ steepest

⇒ will progress towards the $\boxed{\text{steepest}}$ upward progress. → (-4)



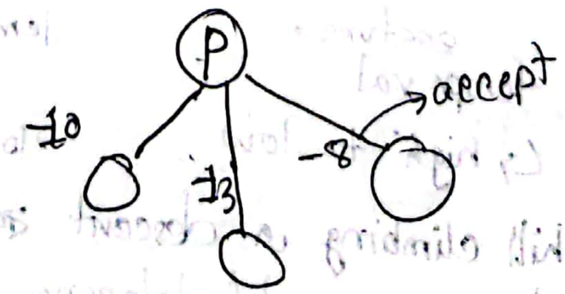
↳ stochastic

⇒ randomly choose one child, except the worst one.



(Faster than the others) ⟶ First choice

↳ first যে child parent এর চেয়ে better.



Genetic Algorithm

Evolution

Survival of the fittest

Population

Person1	Person2	Person3	~~Person4~~

Least fit

1st gen

P1, P2 → offspring 1, offspring 2

P1, P3 → ~~offspring 3~~ ← least fit, offspring 4

2nd gen

offs 2, offs 4 → offs 5, offs 6

offs 2, off 1 → offs 7, offs 8

Goal Generation

Genetic Algo :

⇒ Selection
↓
⇒ Crossover (offspring creation)
↓
⇒ Mutation
↓
⇒ Goal test. (fitness calc)

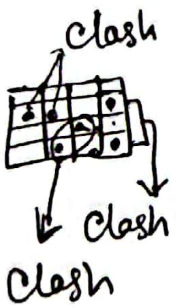
pass ↘ end

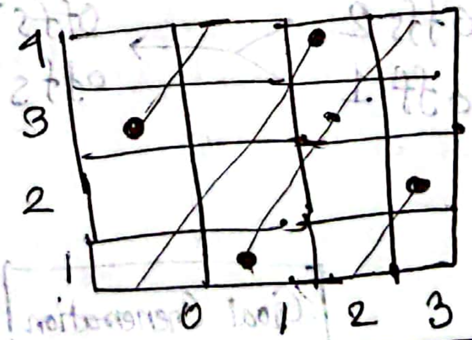
N-Queen

n×n board
↑
n no. of queens needs to be placed

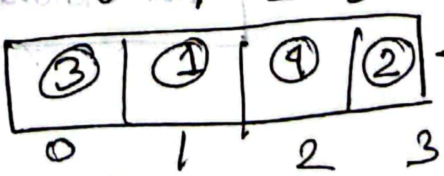
→ No vertical, horizontal or diagonal clash.

no. 4,
4 queen,
4x4 board
4 queens.



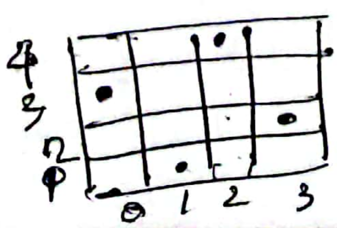


← solution



→ row no, where the queens are

↓



8-queen problem:

board size = 8×8

no. of queens = 8

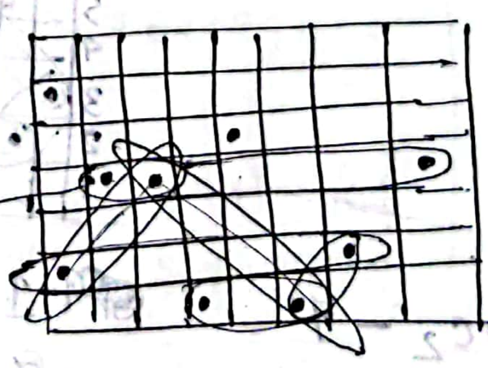
Selection
Crossover
Mutation
Goal test

Phase 1: Selection: 4 8-queen board

24415124 → board 1
32543213 → board 2
32752411 → board 3
24748552 → board 4

Random

board1: 24415124



$^{n}C_2$

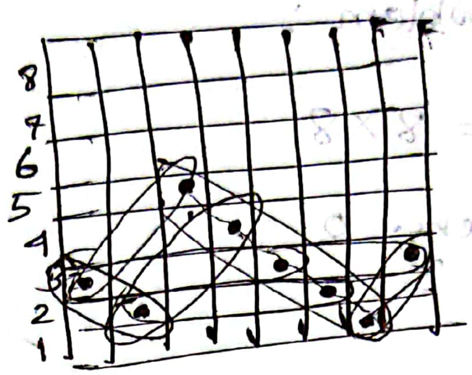
$= {}^{3}C_2 = 3$

$1+1+1+1+1+3$
$+1$
$= 5+3$
$= 8$ pairs

total no. of possible pairs, $^{8}C_2 = 28$,

$\therefore$ fitness $= 28 - 8$
$= 20$ (no. of non-clashing pairs)

32543213



$1+3+1+{}^{5}C_2+1+1$

$= 17$

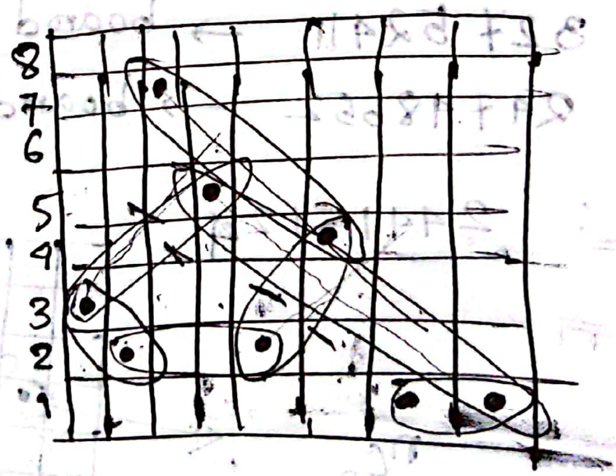
$\text{fitness} = {}^{8}C_2 - 17$

$= 11$

24415124 → 20

32543213 → 11

32752411 → 7



$1+1+1+1+1$

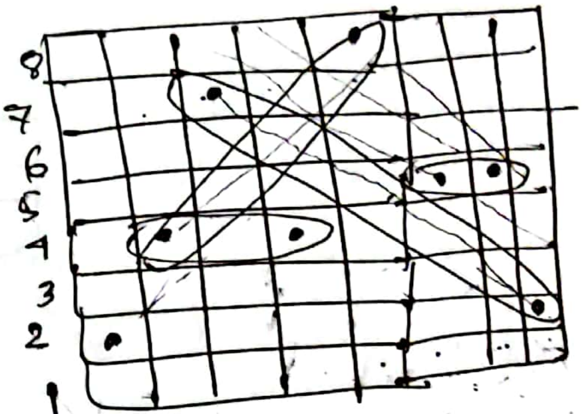
$= 5$

${}^{8}C_2 - 7$

$= 28 - 5$

$= 23$

24748552



fitness = 28 - 4
= 24

1+1+1+1 = 4

Genes
↓

Chromosome → C1

Population

(C1) 24415124 → 20

(C2) 32543213 → (11) ✗ lowest fitness

(C3) 32752411 → 23

(C4) 24748552 → 24

C1 2441 5|124
C3 32752|411 → Crossover → 24415411, 32752124

C1 244|151|24
C4 247|485|52 → Crossover → 24415152, 24748524

Offspring

Mutation: 2 $\boxed{4}$ 4 1 5 4 1 1 → 2 $\boxed{8}$ 4 1 5 4 1 1

~~32752124~~ → 32752124

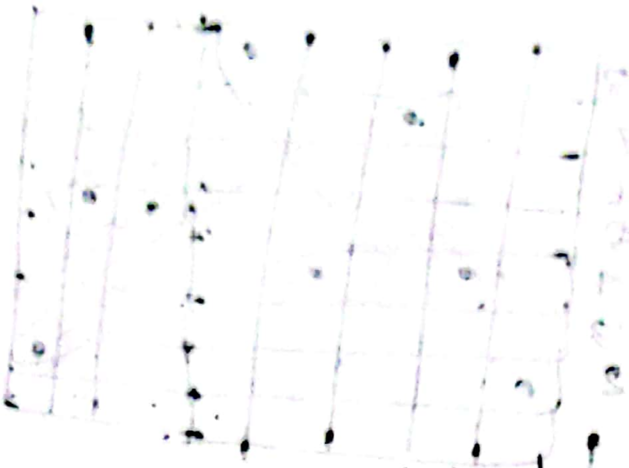
2 4 4 1 5 $\boxed{1}$ 5 2 → 2 4 4 1 5 $\boxed{3}$ 5 2

2 4 7 $\boxed{4}$ 8 5 2 4 → 2 4 7 $\boxed{3}$ 8 5 2 4

↓ fitness calculate

(28 - 0) ⟵ goal test (28)

Scanned with CamScanner



Mutation is needed because,

① offspring দের parent এর তুলনায় diversify করা।
↳ new traits introduce.

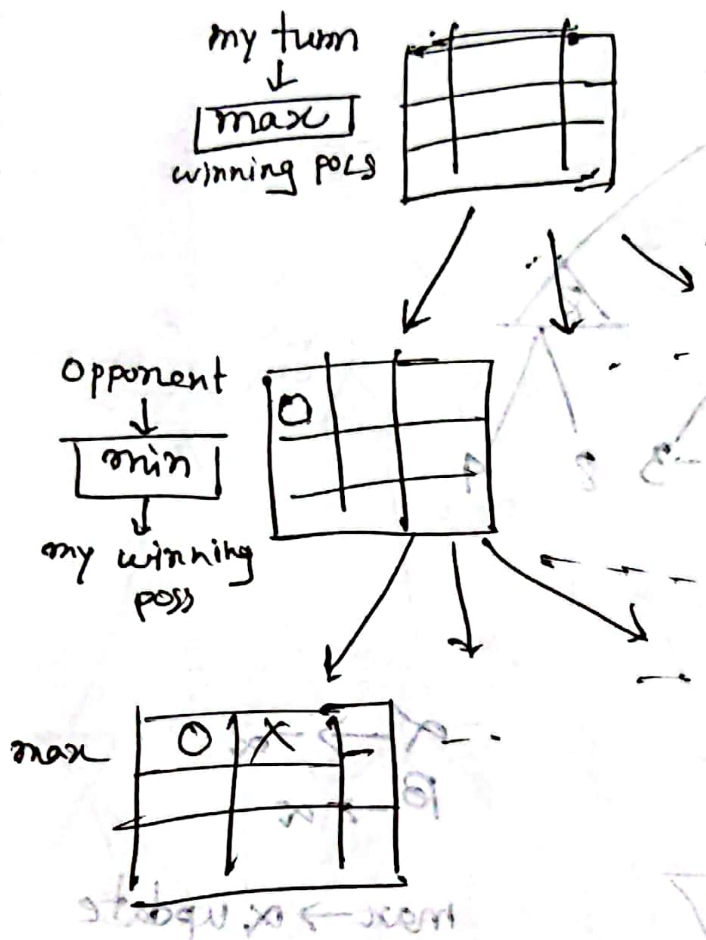
② Parent population এ কোনো trait absent থাকলে, ওই ~~trait~~ trait offspring এ আনার জন্য → mutation.

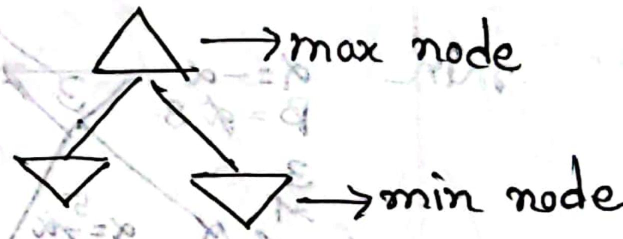
31 4 2

3313 ← no 2
3411
4313
4143 ← dead end

crossover করলেও ২ আসবে না

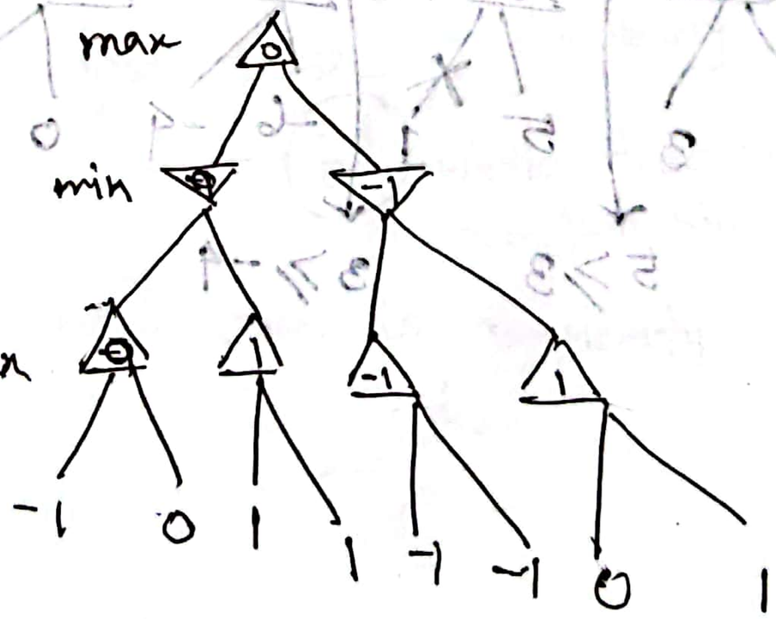
Game





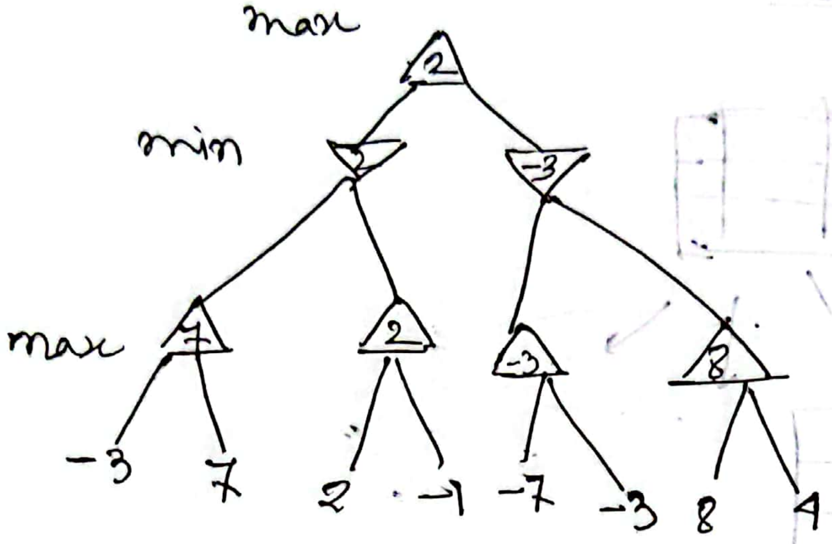
Minimax: for each computer turn

win = 1
lose = -1
draw = 0

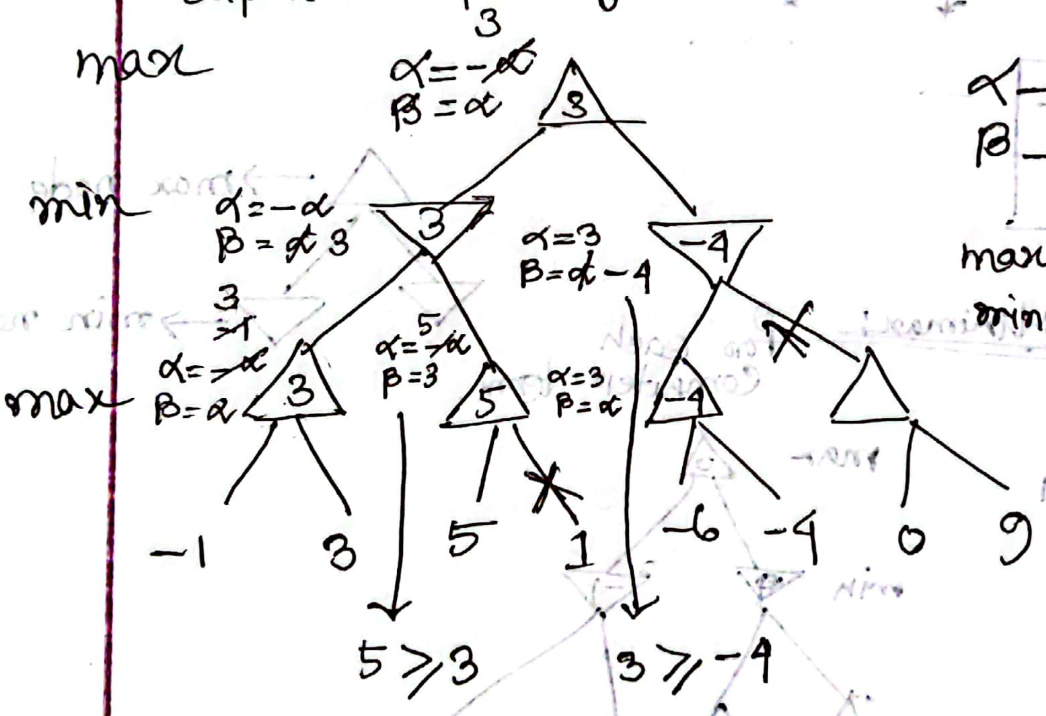


Chess → 10^{12} state space tree.

minimax → computational time বেশী ।



alpha-beta pruning :



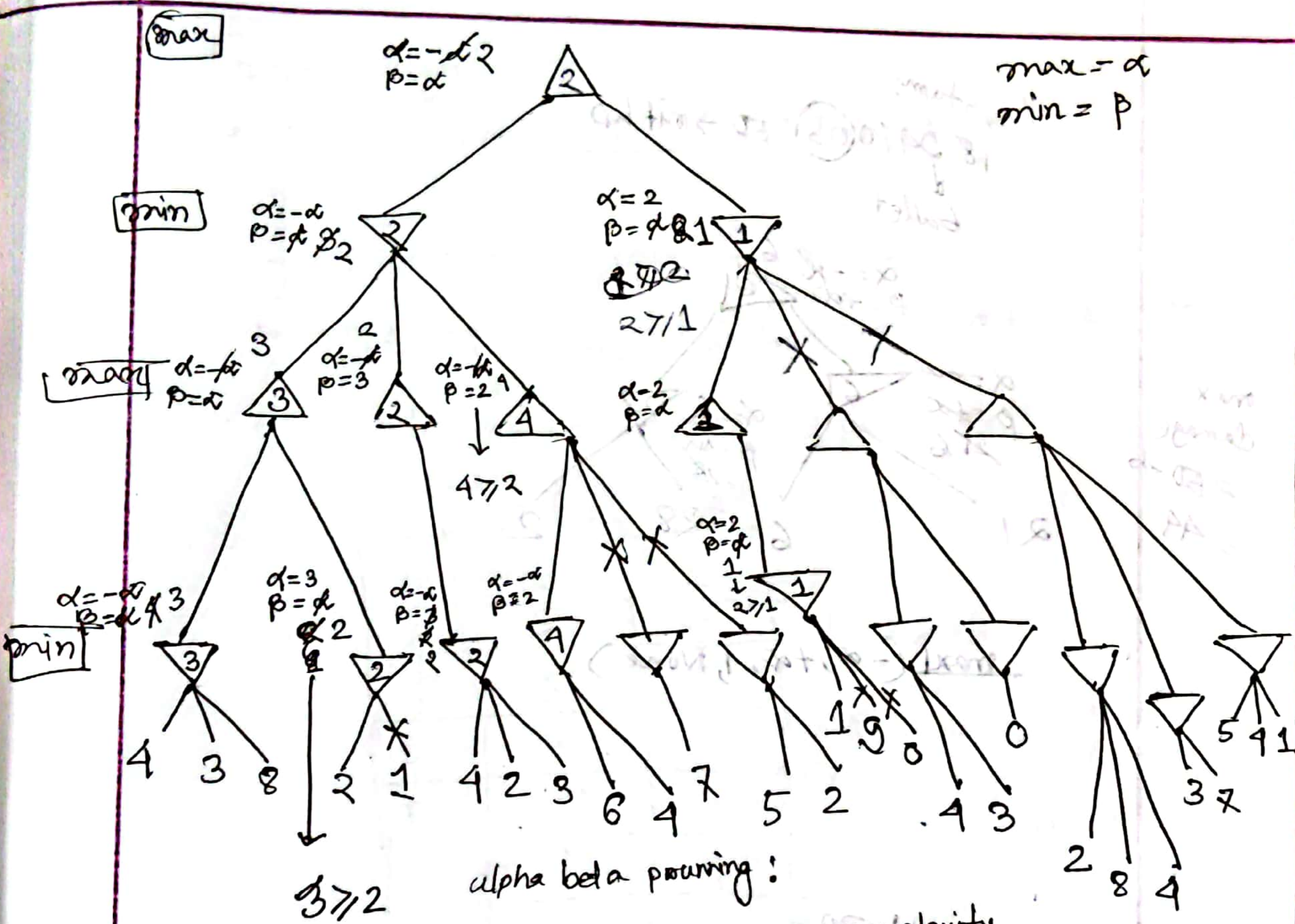
$\alpha \to -\infty$

$\beta \to \infty$

max → α update

min → β "

$\alpha \geq \beta \to$ prune



alpha beta prunning :

worst case ↳ complexity

→ b^d whole tree traverse, no prunning.

best case ↳ complexity,

$(2b)^{d/2}$