(Various Searching)

AI (2092)

* Difference

brute force / blind

Uninformed (blind)	informed / heuristic
i) Search without information.	i) Search with information.
(ii) time Consuming.	ii) Quick Soolution.
(iii) more Complixity	(iii) less complixity.
iv) DFS, BFS	(iv) A*, best first Search
(v) No knowledge	(v) use Knowladge to find.

Serch strategies evaluate in terms four Criteria.

1. Completeness - Solution find out করে কিনা
2. Time Compllxity - how long does it take to find sln.
3. Space complexity - how much memory does it need to perform
4. optimality - If its find the highest quality Solution. when there are several diffirent solution.

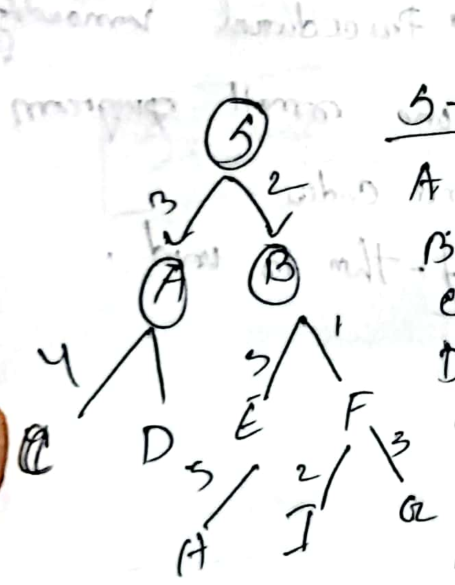
use only heuristic function

Best First Search (Greedy)

use ~~only~~ only heuristic function.

$h(n)$ = cost from node n to goal.

- implemented by priority Queue.



S - Goal	=	13
A - G	=	12
B - G	=	4
C - G	=	7
D - G	=	8
E - G	=	8
F - G	=	2
H - G	=	4
I - G	=	9
G - G	=	0

Initialization : open [A, B], closed [S]

Iteration 1 : open [A], closed [A, B]

Iteration 2 : open [A], closed [S, B]

open [E, A], closed [S, B, F]

Iteration 3 : open [I, G, E, A] closed [S, B, F]

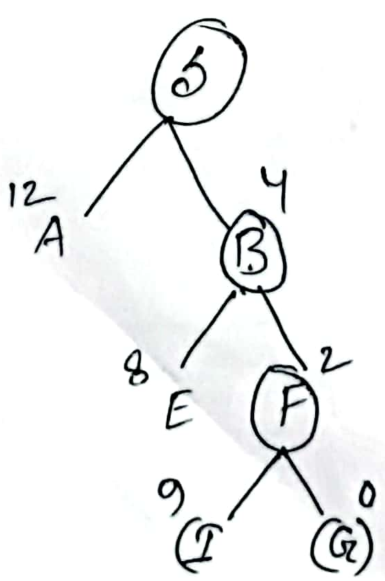
open [I, E, A] closed [S, B, F, G]

প্রতি iteration এ (closed) এ নিতে

open = যেগুলো ভিজিট করা না

closed - করা হয়েছে

ছোট কস্ট ভিজিট দিয়ে যাবো



Path : S → B → F → G

[Complexity $O(b^d)$ }

Unifrom Cost Search · এটাও blind search

8 puzzle problem without heurestic

n এর মান যদি 8 হয় তাহলে $\sqrt{8+1} = 3$ (3×3) Matrix

" 15 " " $\sqrt{15+1} = 4$ (4×4) "

Moves — up, down, left, right

O	X	O
X	$	X
O	X	O

$ = 4 দিকে যাবে

O = 2 " "

X = 3 " "

Qu

S

1	2	3
	4	6
7	5	8

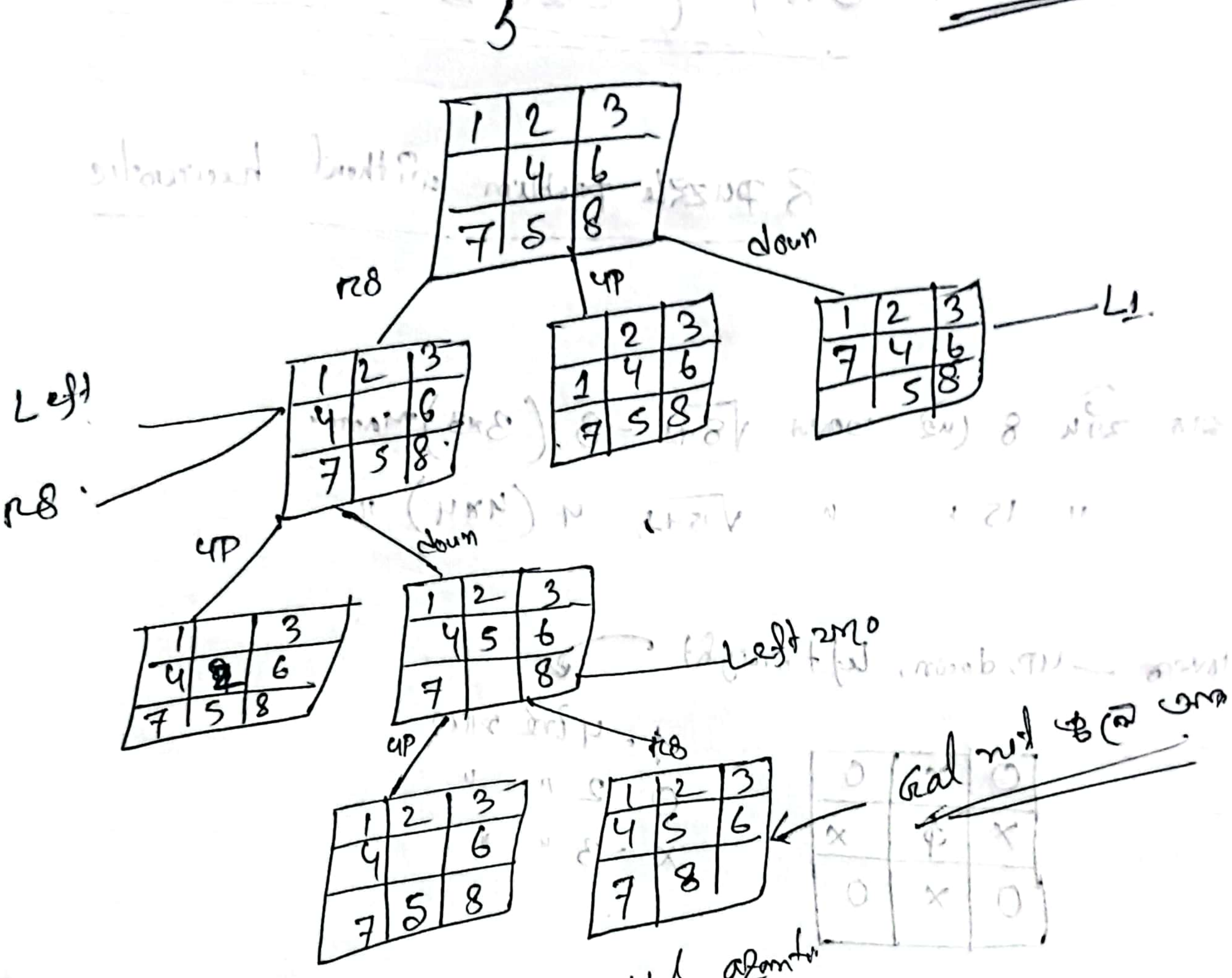
শুধু 8 এর move possible.

কিভাবে আনা হবে →

G

1	2	3
4	5	6
7	8	

BFS দ্বারা করো



Complexity $O\binom{d}{b}$ — কতটি total অবস্থান; কতটি branch গড়ে $\frac{24}{9} \approx 2.6 \approx 3$

এখানে $O\binom{20}{3}$.

8 puzzle problem with heuristic

⇒ optimal ও হতে পারে না

S

1	2	3
	4	6
7	5	8

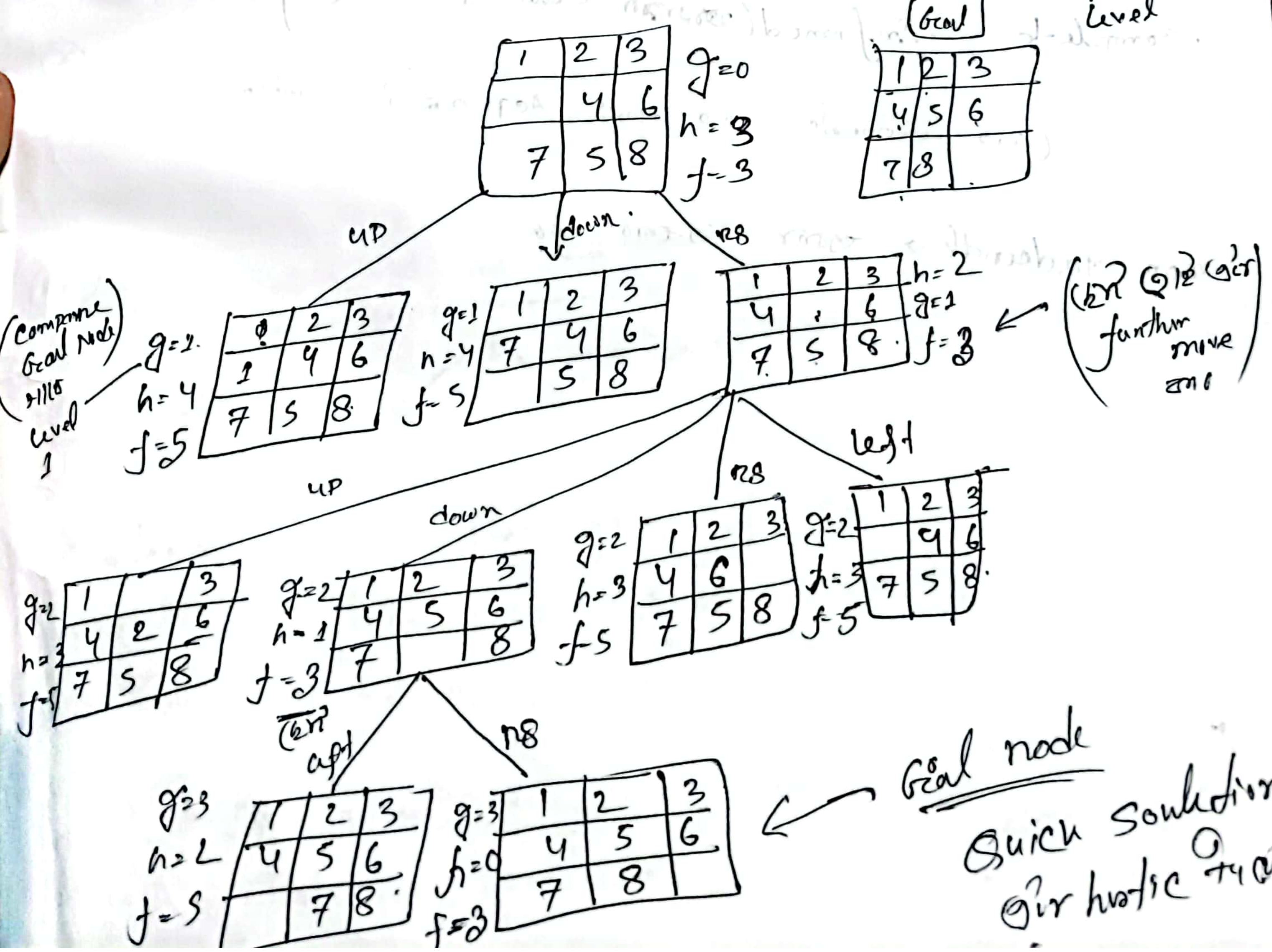
G

1	2	3
4	5	6
7	8	

heuristic value $h = (1 + 1 + 1) = 3$ (St → Goal node 3 Er value নিতে হবে না (4, 5, 8) ঠিক)

$h = 0$ হতে হবে।

cost fun = $f = (G + h)$
G — level



Goal node

Quick Soulution

Heuristic এর জন্য

~~State space Search~~

Heuristic Search → Generate and test Search

heuristic, DFS with backtracking.

1. Generate a possible soln
2. test to see if this a actual soln
3. soln find. Quit. otherwise go to step I

properties of good generator

- Complete
- Informed (প্রত্যেকটি sauln বের করতে পারবে যেন)
(যত Generate হবে test ACP হলে নিবো বাকিগুলো বাদ)

non redundant → একই অবস্থা বারবার

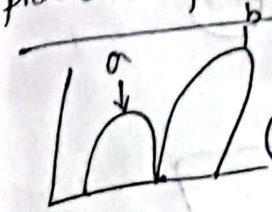
Hill climbing

* better state পাইলে যাইতে থাকবে, backtracking হবে না

Example (8 puzzle), travelling Salesman

(Local Search) - blind নয়

problem of hill climbing

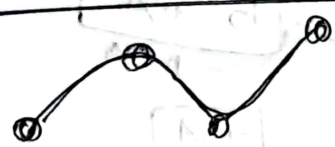


(*) local maximum a থাকলে b তে যাইতে পারবে না।

(*) যদি সব heuristic same থাকে। তখন আর যাবে না (flat maximum)
(যে best move (move গুলো সব এক হয়))

(*) Limitations of Hill climbing

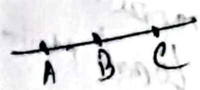
Local Maxima:



← Next state generate হবে না
next page example.

Plateau = is a flat area of the search space in which whole set of neighbouring states have the same value. (সব সমান হলে best আসবে না)

| সব same হলে



Ridge = It is an area of the search space that is higher than the surrounding areas and itself has a slope

Hill climbing (blocks world problem)

(back tracking -হয়)

Local heuristic function: ঠিকঠাক থাকলে +1
ঠিক না থাকলে " -1.

Initial		Goal	
A	-1	D	+1
B	+1	C	+1
C	+1	B	+1
D	-1	A	+1
h = 0 initial		Goal state h = +4	

Maximization problem ⇒ 0 ⇒ 4 যেতে হবে

Initial state.

-1 A
+1 D
+1 C
-1 B
h=0

(heuristic হিসাব.
Goal state নিচে A.
So A থেকে st হলে +1
তবে A উপর B থাকলে +1 আসবে না -1
e মিলবে +1 আসবে না -1)

Goal state.

+1 D
+1 C
+1 B
+1 A
h = 4

এক টাইমে একটি block নাড়াতে পারব

↓

+1 D
+1 C
-1 B
h = ~~0~~ 1

new state.
A +1
+2

⇒

+1 C, B; -1 D, +1 A
h=0

-1 A
+1 D
+1 C
-1 B
h = 0

+1 C
+1 A -1 B -1 D
h=0

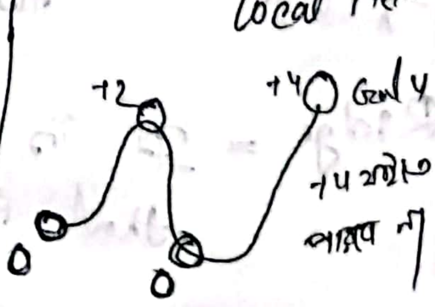
+2

0 0 0

আমার current state, h=0
new " = 2

So এটাই Correct

So এটা trapped in local maxima



Global heuristic function => আগে হচ্ছে যত মিলবে +1 করবে
না মিললে -1 করে করবে (যোগ করে)

-3 A
-2 D
-1 C
-0 B
Initial h => -6

(Maximization problem
(-6 => +6))

D +3
C +2
B +1
A +0
Goal h => +6

নিচে থেকে ABCD এমন ছিল

2টা incorrect
-2 D
একটা incorrect -1 C
তিনটা incorrect -0 B   A 0
নাই তো
0

h = -3
better so dumbo.

-1 C -1 D
0 B 0 A
h = -2

0 D -1 C 0 B A 0
h = -1 Gen better state.
-3 থেকে -1 বেশি so better

-3 A
-2 D
-1 C
-0 B
h = -6

2o current Gen.

D C A
B

অন্যান্য state.

C D B A ← Gen better state -1 থেকে বেশি
0 0 0 0

+1 B
0 C 0 D 0 A
+1

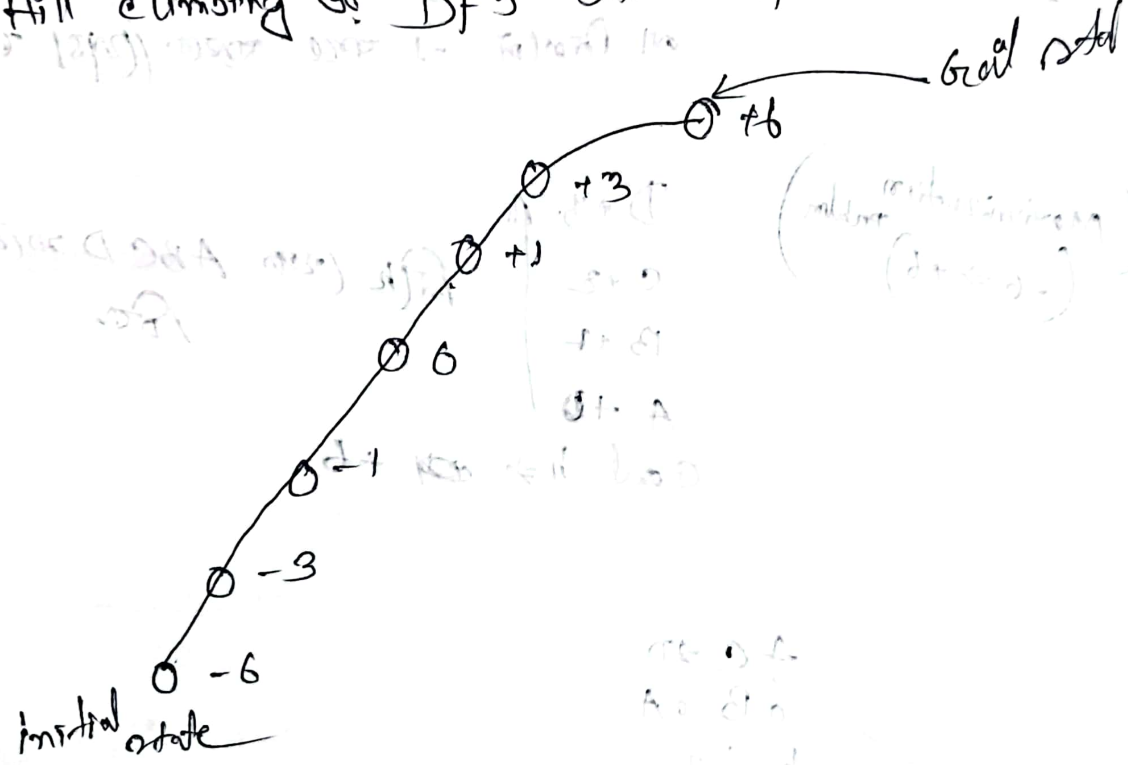
→

+2 C
+1 B
0 D 0 A
= 3

→

+3 D
+2 C
+1 B
0 A
h = 6.
= 6 ← Goal state

Hill climbing Go! DFS state space tree



A* Search Algorithm

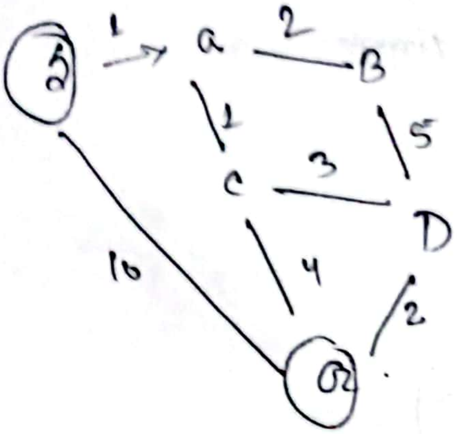
কোন এক জায়গা থেকে আরেক জায়গায় যেতে cost

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

$f(n)$: fitness number

$g(n)$: cost to reach node n from start

$h(n)$: Cost to reach goal from node n.



n	h(n)
S	5
A	3
B	4
C	2
D	6
G	0 ← goal

Soln

$f(S) = g(S) + h(S)$
$= 0 + 5 = 5$

$S \rightarrow A$

$f(A) = g(A) + h(A)$
$= 1 + 3$
$= 4$ minimum

$S \rightarrow G$

$f(n) = g(n) + h(n)$
$= 10 + 0$
$= 10$

$S \rightarrow A \rightarrow B$

$f(n) = g(n) + h(n)$
$= (1+2) + 4$
$= 7$

$S \rightarrow A \rightarrow C$

$f(n) = g(n) + h(n)$
$= (1+1) + 2$
$= 4$

Minimum cost Chose করা হয়

$S \to A \to C \gg \begin{matrix} G \\ D \end{matrix}$ বাদে

এভাবে যেগুলো গেলো না সেইগুলো Last এ বাদ

$S \to A \to C \to D$

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

$= (1 + 1 + 3) + 6$

$= 11$

$S \to A \to C \to G$

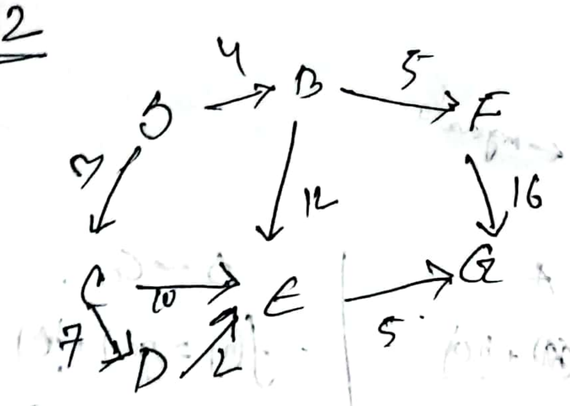
$f(s) = g(n) + h(n)$

$= (1 + 1 + 4) + 0$

$= 6$ Minimum

$\boxed{S \to A \to C \to G}$

Ex 2



(n)	h(n)
S	14
B	12
C	11
D	6
E	4
F	11
G	0

$S \to B$

$4 + 12 = 16$

$S \to C$

$3 + 11 = 14$

$S \to C \to D$

$= (3 + 7) + 6$

$= 16$

$S \to C \to E$

$3 + 10 + 4$

$= 17$

$S \to B = 1$ বাদে

$= 16$

হইলে same হইলে calculate

S → C → D → E

$(3+7+2)+4$

$= 16$ এটা optimal.

S → B → E

$4+12+4$

$= 20$

S → B → F

$= 4+5+11$

$= 20$

S → C → E

$= 17$

এগুলো থেকেও এগুলো বেশি so বাদ

S → C → D → E → G

$(3+7+2+5)+0$

$= 17$ ← এটা minimum আসছে

Path → S → C → D → E → G

A* Admissibility

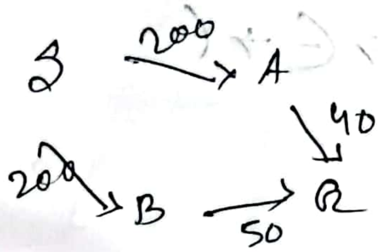
↓ gerante Guo a optimal Sul^n Gom. Pafo

Underestimation h^* $h(n) \leq h^*(n)$

Estimate cost — actual cost.

Overestimation. $h n \geq h^*(n)$

Ex



n	h(n)
S	5
A	
B	
G	0

Case 1 (overestimation.)

let $h(A) = 80$.
$h(B) = 70$

$S \to A$	$S \to B$
$200 + 80$	$200 + 70$
$= 280$	$= 270$

$\Rightarrow$ $S \to B \to G$

$200 + 50 + 0$

$= 250$ optimal.

Case 2 Underestimation

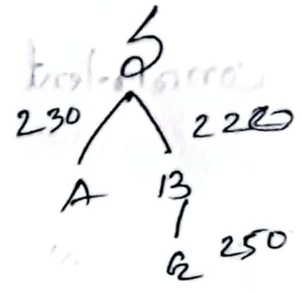
Let $h(A) = 30$
$h(B) = 20$

$S \to A$ 230

$S \to B$ 220 >> $S \to B \to G$
$200 + 50 + 0 = 250$

$S \to A \to G$
$200 + 40 + 0$
$= 240$

$S \to A \to G$ (underestimation Guarantee দেয়)



A* search always optimal দেয় না

(i) $h(n) > $ actual cost
$h(n) = $ actual cost (best)
$h(n) < $ actual cost ← তখন সমস্যা হবে

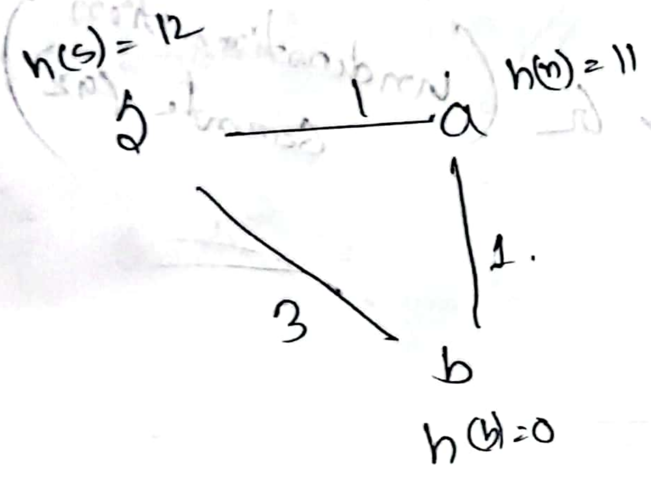
যদি $h(n) \leq$ actual cost
হবে admissible heuristic

যদি S → a, h(n) 12, h(a) = 11
যদি b (12) ≤ 1 + 11
then constant heuristic
↓ হলে optimal soln হবে

(খ) Consistent হলে of course admissible
বা optimal দিবে।

(গ) u না হলে optimal দিবে না

Consistent finding



যেহেতু

$s \rightarrow A$ এর

জন্য

$h(s) \leq cost + h(a)$

$= 12 \leq 1 + 11$

$= 12 \leq 12$

So এর consistent heuristic

$a\ s \rightarrow b$

$h(s) \leq cost + h(b)$

$12 \leq 3 + 0$

$\boxed{12 \leq 3}$

মানে না

So এর Consistent না

optimal দিবে না

Agent / Intelegent Agent

Percept (current + history) (shirla কর).

Environment → Agent program

Effectors
(action perform কর.

[Agent → percept → Decision → Actions.] ⇒. যার environment Change কর।

Goal of Agent → high performance.
→ optimized result
→ Rational Action

<u>Agent design model</u> :> কোনো agent বানাইতে চাইলে ওগুলো rhate হবে.

P - Performance — Saffty, time legal drive

E - Environment → roads, other vehicles, sign

A → Action, → Stearing, accelerator, brake.
Actuators.

S → Sensors → Camera, GPS, speedmeter

Example of PEAS

Agent : Part-picking Robot

Performance measure : Percentage of parts incorrect bins

Environment : Conveyor belt with parts, bins.

Actuators/action : jointed arm and hand

Sensors : Camera, joint, angle sensors.

Environment types

Fully observable vs partially observable

- → agent এর sensor এর সাহায্যে fully access করতে
- → internal state trax rahke হবে

environment partially observable হয় যদি sensor noisy and inaccurate হয় or parts of the state missing from the sensor data

Ex: Vacuum cleaner with local dirt sensor

taxi driver

Deterministic vs Stochastic

- Current state ও বলে Next state জানতে পারলে
- Deterministic full observable হলে don't worry ~~কোনো~~

আর partially observable হলে Stochastic

Ex: Vacum world is deterministic, taxi is not deterministic না

Episodic

- agent experience is divided into atomic "episodes"
  Example: classification tasks, part picking robot.

Sequential: যখন Current decision affect করে all future decision কে

Ex: Chess and taxi drive

Single agent

যদি কোন agent involved থাকে (operating by itself)

Ex: Maze

Multi agent: multiple agent operate

Ex: football.

Static: Environment change হয় না যদিও agent acting থাকে.

Ex: Crossword puzzles

Chess board ও

Dynamic: change over a time

Ex: roller coaster, taxi driving

সময়ের সাথে সাথে change

Discrete: finite number of action that can be deliberated

কোন output পাওয়া যায়

Ex - chess

Continuous: Performance can't numbered করা যায়

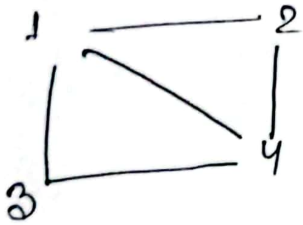
Ex: taxi driver

CamScanner

Constraint Satisfaction prbl (CSP)

Backtracking এর মত

এখানে পাশাপাশি
same color হবে না



$V = \{1, 2, 3, 4\}$

Domain = {Red, green, blue}

Constraint = $(1 \neq 2)$ $(1 \neq 3)$ $(1 \neq 4)$ $(2 \neq 4)$ $(3 \neq 4)$

	1	2	3	4
Initial Domain	RgB	RgB	rgb	rgb
1 = R	R	G,B	G,B	GB
2 = G	R	G	GB	B
3 = B	R	G	BG	B

2, 3 Neighbour না so same
কালার হয়

Genetic Algorithm

- tool biological Evolution
- form an optimization, Complex (NP, hard) problem solve করে
- From group of individual, the best will survive.

$$\text{Phenotype} \xrightarrow{\text{Encode}} \text{Genotype} \xrightarrow{\text{decode}} \text{Phenotype}$$

initial population
↓
Calculate fitness ⟷ abcde / fghi ⟩ 2 (a=a) মিলিয়ে দেখে fitness.
↓
Selection → ২০ বেশি fitness ৩০ বেশি selection. ৩০ বেশি fitness select করে
↓
Crossover → swap করে যেই যাদের fitness বেশি
abcd / abfg ⇒ abfg / abcd.
↓
Mutation → some fitness better করে
↓
Stopping criteria. — No → Calculate fitness
↓ yes
optimum

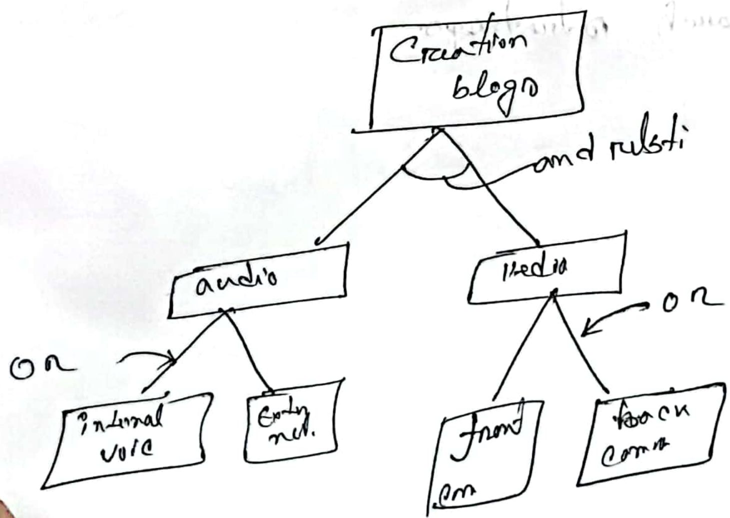
Problem Reduction

Complex problem
- Devide sub problems
- Combine করে Ans বের করা

relationship = And

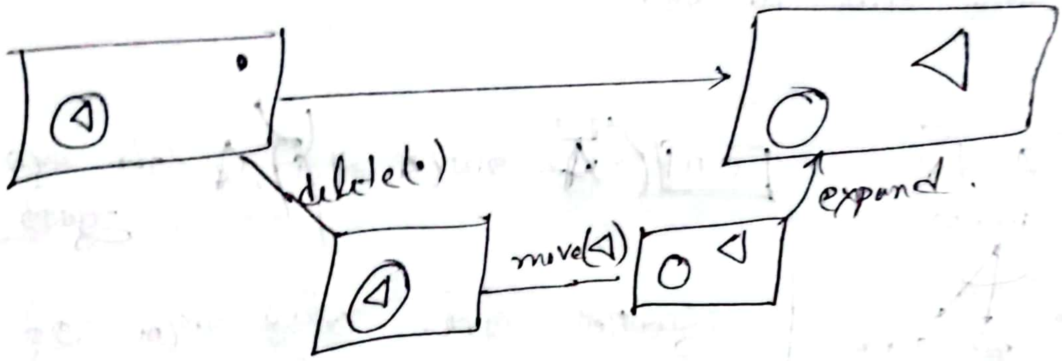
OR (A অথবা B বা C) A এর মানে And বুঝায়

যেকোনো একটার solve করলে হয়ে যায়



(MEA) Means ends analysis — Problem solved technique.

initial state.



Sub problem Solve করে বড় Problem Solve করে.

(2) (backward + forward) Search strategies.

Heuristic search

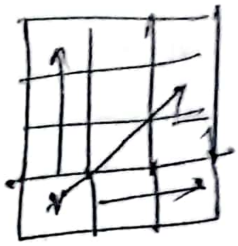
• It is a problem solving technique that uses estimates of how close given state is to the goal state called heuristic. In AI this search teeniqes used for find optimal sulin in complex state space.

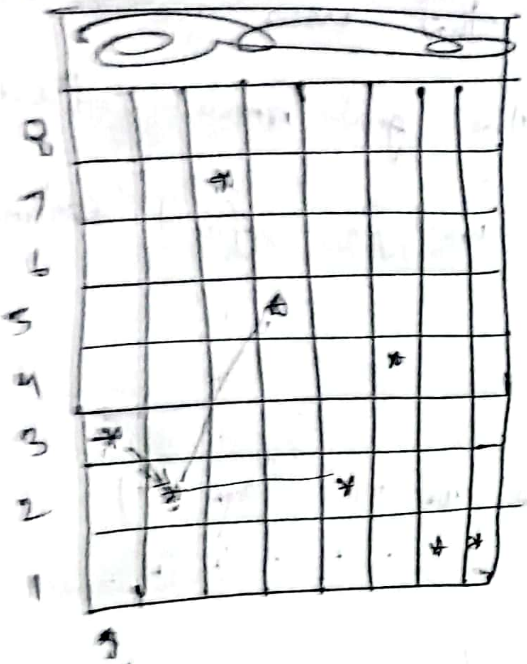
8 puzzle (given theke goal যাইতে কেমন cost লাগে হিসাব করে)

rationally - agent which does the rig thing.

8 Queen Problem

তোমার Queen বসাতে হবে এমন যেগুলো কাটবে না





Column ধরে বসানো

Population

3	2	7	5	2	4	1	1

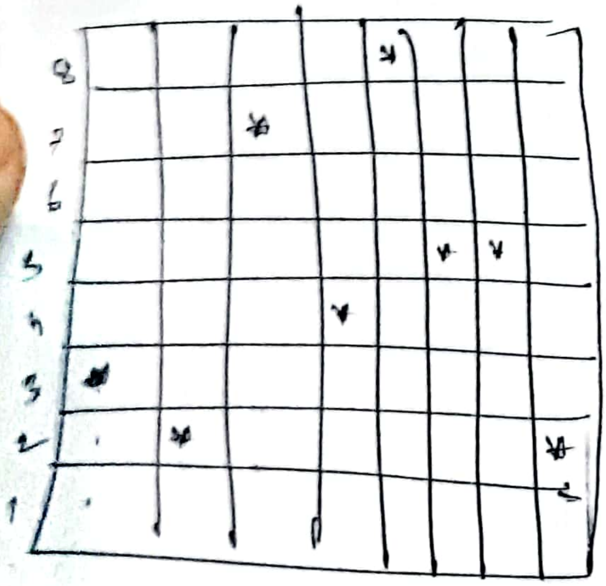
fitness = no of non attacking pairs.

$Q_1 = 6$ $Q_4 = 3$ $Q_7 = 0$

$Q_2 = 5$ $Q_5 = 3$ $Q_8 = 0$

$Q_3 = 4$ $Q_6 = 2$

total = 23



initial population ← (চেঞ্জ করো)

3	2	7	4	8	5	~~5~~	2	~~1~~

fitness = 23

Probability of population $= (23)+(23) = 46$.

$$1st = \frac{23}{46}, \frac{23}{46}$$

$$= 50\%, 50\%$$

Crossover | যে দুইটা ছিল সেগুলো দিলাম

Fitness func.		Selection	Crossover
32752411	23	50% 3274 | 8522	3274 2411
32748552	23	50% 3275 | 2411	32758552

→ 3274 2 411

→ 3275 8 552

mutation.

যেকোনো একটির value Chang করে দেওয়া।

Au23

1(c) or

(block world problem)

Initial state

A -1
H +1
G -1
F +1
E +1
D +1
C +1
B -1

score = 4 (6-2)

→

Goal state

H 1
G 1
F 1
E 1
D 1
C 1
B +1
A +1

score = 8

(c)

Local → ঠিক হলে +1, wrong হলে -1.

Global → 0 থেকে start sequential হলে +1 না হলে -1 (increase হবে)

(ii) Local

-1 A
1 H
1 G
1 F
1 E (4)
+1 D
+1 C
-1 B

⇒

H 1
G 1
F 1
E 1
D 1
C 1
B -1 A +1

(6)

1 G
1 F
1 E
1 D
1 C
-1 B A +1 H -1

(4)

+1 G
+1 F
+1 E
+1 D
+1 C H -1
-1 B A +1

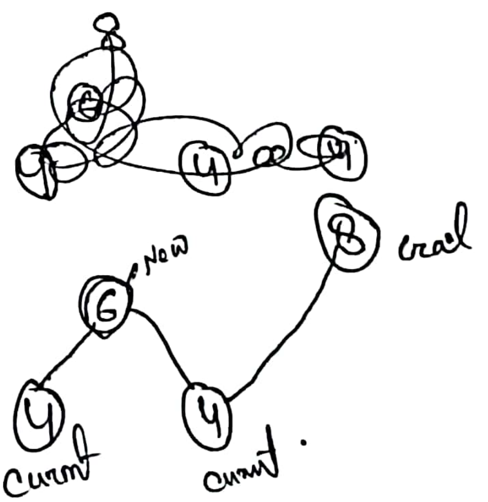
(4)

-1 A
1 H
1 G
1 F
1 E
1 D
1 C

(4)

So in the local heuristic.

Both are (4) ← so there



Local Maximum occured

So we can use global heuristic function.

Start.

A -7.
H -6
G -5
F -4
E -3 (-28)
D -2
C -1
B 0

Goal.

H 7.
G 6
F 5
E 4 (28)
D 3
C 2
B 1
A 0

H -6
G -5
F -4
E -3 (-21)
D -2
C -1
B 0 A 0

→

-5 G
-4 F (-15)
-3 E.
-2 D
-1 C
0 B H 0 A 0

→

F -4
E -3
D -2 (-10)
C -1
B 0 A 0 H 0 G 0

→

E -3
D -2 (-6)
C -1
B 0 A 0 H 0 G 0 F 0

↓

D -2
C -1 (-3).
B 0 A 0 H 0 G 0 F 0 E 0

↓

C -1 (-1)
B 0 A 0 H 0 G 0 F 0 E 0 D 0

⇓ (0)

B 0 A 0 H 0 G 0 F 0 E 0 D 0 C

⇐

B
A

⇐

C
B
A

⇐

D
C
B
A

⇐

E
D
C
B
A

⇐

F
E
D
C
B
A

⇐

G
F
E
D
C
B
A

⇐

7 H
6 G
5 F
4 E (28)
3 D
2 C
1 B
0 A

Goal state.