UNIT-IV

Game playing and current trends in AI

↳ fundamental area in AI reasearch.

#current trends -

- **Reinforcement learning** -
  - → trial & Error ke through play kia jata hai.
  - → learn karta hai actions & consequences se.

- **Generative adversarial Networks** -
  - → used to genrate content with in Games.

[create karta hai Realistic textures, characters & environment]

- **Procedural content Genration** -
  - → content ko dynamically levels, maps & quests jo ki help karte hai player ko replay karne mein

- **player behavior prediction** -
  - → used to predict player move/ behavior.

- **Real time strategy Games**

Games Required long term planning, resource management and quick decision making.

MAX-MIN

→ Recursive, Backtracking

→ optimal move provide karti hai, assume karte hai sarnne wala player bhi optimally khel raha hai.

→ used in two players game (tic-tac-toe, chess)

→ computes min-max decision of current state

→ Procedure -

2 players hote hai ek max, ek min dono fight karte hai, opponent ko ~~maximum~~ minimum benefit milta hai, jabki dusre ko maximum milta hai.

- Max player maximized value choose /select karta hai
- Min player Minimized.

→ Debth first Search use hoti hai Exploration ke lie.

→ terminal node tk jane ke Baad Backtrack karti hai as Recursion.

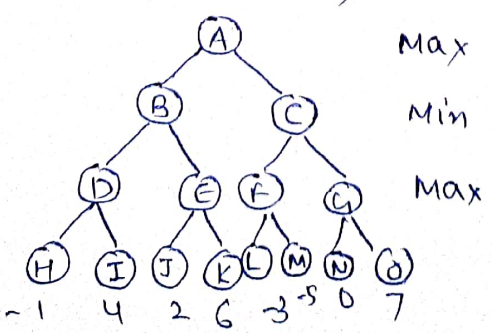
working -

algorithm mein DFS apply karte hai terminal node tak jate hai, terminal node ki value compare karenge max min choose karenge or Backtrack karenge jb tk root node nhi aa jati same process karenge.

Steps

① Generate game tree or utility function bnao utility value ke lie

Maximizer first turn lega worst case ke sath (-∞) value and Minimizer - best (+∞)



② Utility value nikalenge D ke lie Max (-1, -∞)
Max (-1, 4) = 4

E ke lie = (2, ∞) = Max (2, 6) = 6

F (-3, -∞) = Max (-3, -5) = -3

G (0, -∞) = Max (0, 7) = 7

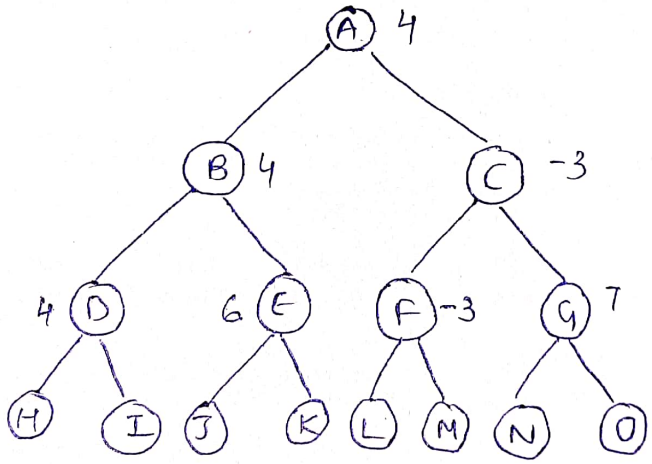
③ ab Min ke lie nikalenge utilities value (B, C) ke lie.

For B = min (4, 6) = 4

For C = min (-3, 7) = -3

④ Phir se max ki turn ayegi ~~A~~ for node A

~~A for~~ For A = max (4, -3) = 4



Properties -

- complete ✓ agr solution Exist Karta hai to zarror mileg ✓
- Backtracking
- optimal
- Recursive
- time complexity = $O(b^m)$

Limitations -

- Slow.

Alpha Beta pruning -

↳ Modified version of Min-Max algorithm.

↳ Bina har node ko check kare, correct Min max decision lena "pruning"

↳ is technique mein threshold parameter hote hai
α, β

α → higher value {Maximizer ka kam karta hai} (-∞)
β → lowest value {Minimizer ka kaam karta hai} (-∞)

↳ Remove kar deta hai un nodes ko jo final Result ko affect nhi karti.

↳ condition for pruning $\alpha \geq \beta$

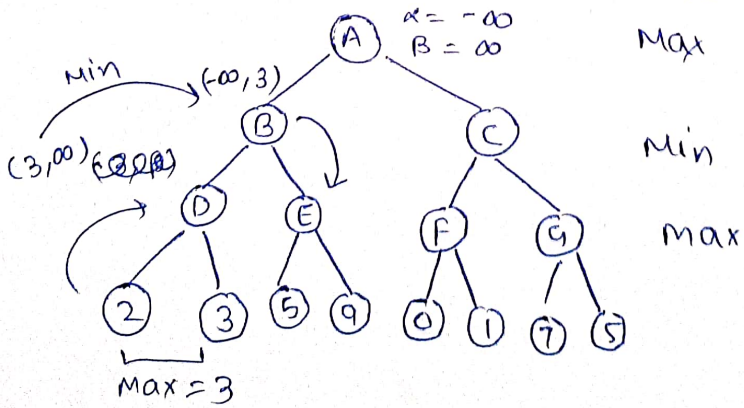
- α ki value sirf Max player update karega.
- β ki value sirf Min player update karega
- Back track karte samay, node values upper node ko pass hongi {α, β ko nhi}
- alpha, beta ki values child node ko di jayengi

Working

Steps

① Max player A node start karega jaha $\alpha = -\infty$, $\alpha = \infty$ yeh value α & β ki pass hongi (B) ko jaha pe $\alpha = -\infty$, $\beta = +\infty$ or no node B, child D ko dega.

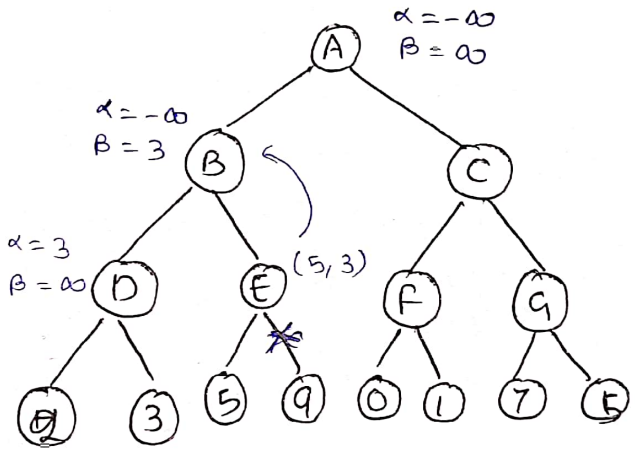
② Node D, α ki value calculate karenge, Max ki turn leke α ko compare karenge 2 se, 3 se or max value nikalenge = 3 to $\alpha = 3$ for D & node value will also 3



③ Backtrack karke node B jana pe min ki value find karenge means β ki {β = +∞ . Compare karenge Min(3, ∞) Min(3, ∞) = 3} so, B node pe value & β ki value 3 ho jayegi

Now, usko E pe pass karenge, matlab α = -∞, β = 3 E ko pass karenge.

④ E pr Max ki turn hai, islie α ki value change hogi, α ki value Max(-∞, 5) = 5 so, (α = 5, β = 3), but ab α ≯ 3 to bachi hui node E (matlab child node E ki jo bachi hai prune kar denge.



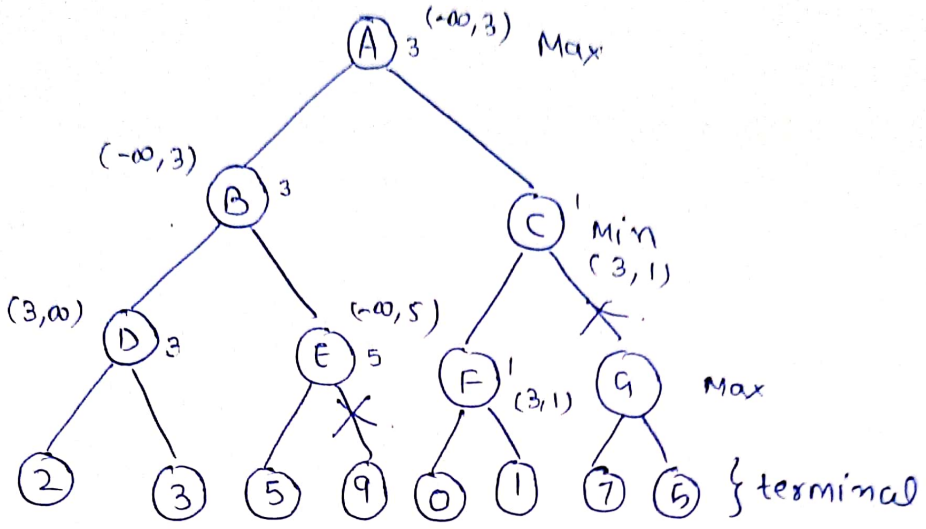
⑤ Backtrack karega, A pr jayega vapis Max ki turn ayega, α ki value change hogi Max(-∞, 3) = 3 β = +∞ yeh value node A se C pr jayegi. & then F pe

At F, α ki value change hogi kaunki Max play karega

Max(3, 0) = 3 & uske baad Rights se compare karenge max(3, 1) = 3 but F node ki value 1 hogi kaunki (0, 1) = 1 ↳ Max.

⑥ Node F 1 value pass karega C ko, α = 3, β = +∞, β ki value change hogi kaunki is baar min play karega.

min(∞, 1) = 1, at C = (α = 3, β = 1) gain condition satisfy kar rhi hai α ≯ β so, G node ko prune kar denge



Significance

- Powerful optimization technique used in Game tree such as Chess, Poker
- Efficiency Improvement
- optimality.
- Branch pruning
- Space complexity ↓ time complexity ↓
- Application in Game playing.

Game Development using AI

(1) **Adaptive difficulty** - adjusting difficultty level, optimal level, Skill level.

(2) **player behavior prediction** - analyze karta hai player ko behavior next move kis pattern se khel rana hai.

(3) **Emotion Recognition & Response** - Environment ke hissab se pta lagana what will be the next move.

(4) **Pathfinding & Navigation** - help characts to find optimal path & moves & avoid obstacles.

(5) **Machine learning for Game testing** - used to automate testing process, optimizing game performance, Ensure better Quality assurance.

Types of Games

aspect	Deterministic	Non Deterministic
EX	tic-tac toe, chess	Card games, Poker
outcome nature	outcome pta chl jata hai player ke moves ke Predictions se	uncertainity hoti hai kon jeetega
strategy	Perfect information & Player can decide their move	incomplete information adaption to uncertainity
Complexity	Strategies can be deeply analyzed	complexity arrises from inherent
Info availability	complete info about game	Info is hidden & uncertain