Date _____ Day _____

"CHAPTER #01"

"Artificial Intelligence is the science & engineering of making intelligent machines, especially intelligent computer programs"

"Intelligence is the computational part of the ability to achieve goals in the world"

"AI is the study of agents the recieve precepts from the enviornment & perform actions

→ Rationality is
1. Maximally achieving pre-defined goals
2. Goals are expressed in terms of the utility of outcomes

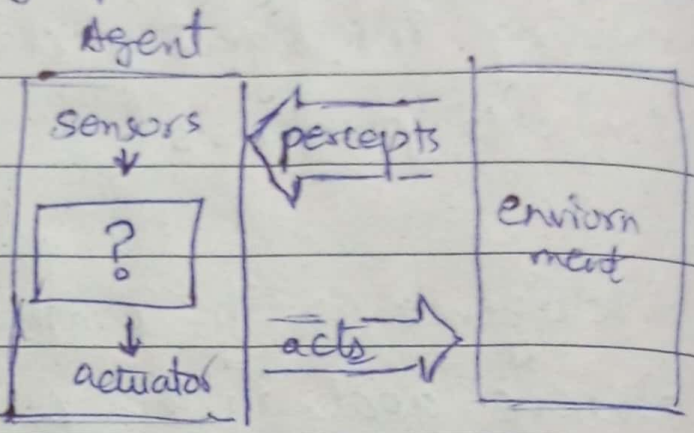
• Turing Test

→ A Computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a computer

Date _____ Day _____

o Agents

An agent is anything that can be viewed as perceiving the env through sensors and acting upon that env through actuators.



- percept refers to the content an agent's sensor are perceiving
- percept sequence is the complete history of everything the agent has ever perceived
- An agent's behaviour is described by the "agent function" that maps any given percept sequence to action

→ A rational agent is one that does the right thing

→ Doing actions in order to modify future percepts - called information gathering

→ An agent is autonomous if its behaviour is determined by its own experience (with ability to learn & adapt)

Date _____ Day _____

PEAS

- Performance Measure:
  - Parameters or metrics on which the performance of the agent would be analyzed
- Environment:
  - An env the agent will be working on.
- Actuators:
  - Device responsible for performing actions
- Sensors:
  - Device used to perceive.

Interactive English Tutor Agent:

- P: Student's score on test
- E: Set of students, testing agency
- A: Display of excercises, feedback, speech
- S: Keyboard entry, voice

Date _____ Day _____

▷ ENVIRONMENT TYPES

① Fully Observable Vs Partially Observable

↳ In fully observable the agent has complete & accurate info about the env at any given time. Like in Chess. where entire board state is available

↳ In Partially observable the agen has limited or incomplete info about the env due to missing data, noise or constraints. Like in Cards where player cannot se other's cards.

↳ if agent has no sensors at all then env is unobservable. Like robo navigating in darkness

② Deterministic Vs Stochastic (non deterministic)

↳ In deterministic the next state of the env is completely determined by the current state and the agent's action. Eg: Chess

↳ Non deterministic the next state has some degree of randomness meaning the same action can lead to diff outcomes. Eg: Rolling a dice

Date _____ Day _____

③ Episodic Vs Sequential (non-episodic)

↳ An agent experience is divided into atomic episodes

↳↳ The agent's action in one episode donot effect the future episodes. Each decision is independent of past actions. Eg: Image classification

↳ In sequential the current actions influence future states and rewards. Eg: Chess, Autonomus car.

④ Static Vs Dynamic

↳ In static the env doesnot change while the agent is deciding on an action. Eg: Chess

↳ In dynamic the env changes over time, even if agent does nothing, requiring real-time decision making. Eg: Self driving car

⑤ Discrete Vs Continuous:

↳ In discrete there are limited no. of distinct, clearly defined percepts & actions. Eg: Chess

↳ In continuous the env has an infinite no. of possible states & actions. Eg: A robo moving in 3D space

Known & The agent knows the rules of the env, like in game of chess
Unknown: The agent doesn't fully understand rules of env and must learn

Date _____ Day _____

⑥ Single Agent Vs Multi Agent

↳ In single agent only one agent interacts with env.
Eg: Sudoku Solver

↳ In Multi Agent, agent interact with multiple agents.
Eg: Chess

▷ TYPES OF AGENTS

① Simple Reflex Agents:
- Acts based on current percept only (ignoring history)
- Use if-then else statements
- Eg: A thermostate turning on/off based on temp.

② Model Based Reflex Agent:
- Maintain an internal model of env
  ↳ it is the knowledge about how things happen in the world
- Consider past states to make better decisions.
- Eg: Self Driving car

Date _____ Day _____

③ Goal Based Agents:

- ~~Optimize actions based~~ Act to achieve specific goal rather than just reacting
- Involves consideration of the previous & future
- Requires planning and decision making
- Eg: A GPS system finding the best route to dest

④ Utility Based Agent:

- Optimize actions based on a utility function
- Utility func to decide which state is better for agent
- They find best way to achieve the goal.
- The utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action
- Eg: A stock trading AI maximizing profit while minimizing risk

⑤ Learning Agents:

- Improve their ~~act~~ performance over time by learning from experience
- Use ML techniques to adapt

Date _____ Day _____

- Four conceptual components
  ① Learning element ② Critic ③ Performance Measure
  ④ Problem generator
- Eg: Recommendation System

▷ Problem Formulation:

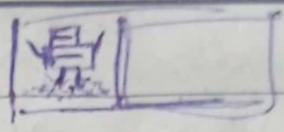
→ Problem Spaces States Space

(1) State: Set of all possible states where you can be
(2) Initial State: The state from where search begins
(3) Actions: All the possible actions agent can perform.
→ ACTION(s) returns finite set of action that can be performed on s
(4) Transition Model: Which describes what each action does.
RESULT(s, a) returns the state that results from doing action(a) on states(s)
(5) Action Cost Function: ACTION-Cost(s, a, s') gives numeric cost of applying a on s to reach s'

→ A sequence of action forms a path
→ A solution is a path from the initial → goal state
→ An optimal solution has the lowest path cost among all solutions

Date _____ Day _____

Vaccum World:



(1) States: Too few cells
(2) Initial States: Any state can be designated as initial state
(3) Actions: Suck, moveLeft, moveRight
(4) Transition Model: Suck removes dirt. Move right & left change the direction by 90°
(5) Goal States: The states in which every cell is clean
(6) Action Cost: Each action costs 1

depth of least cost solution

"SEARCH STRATEGIES" Time Complexity (b^d)

b max branching factor of search tree

Blind Search / Uninformed Search	Informed Search / heuristic search
• Searches without any prior knowledge (such as closness or location of goal)	• Use knowledge such as heuristics, to guide the search
• Time consuming as it explores all possible paths	• Lesss time consuming
• DFS, BFS, Uniform Cost Search	• A*, Heuristic DFS, Heur BFS, Best-First Search, Hill climbing
• No. of steps, path cost unknown	
• agent knows when it reaches a goal	• Heuristic is a function that estimate costs or distance from current state to goal state.
• Operates in a brute force manner	

Date _____ Day _____

▷ Breadth First Search :

Completness : Yes, a solution will be found if exists

Time Complexity : $O(b^{d+1})$

Optimal : Yes.

→ Suppose the branching factor $b=3$, and the goal is at depth $d=20$ then, time to finish $= 3^{20}$

→ Not suitable for large graphs

▷ Depth First Search :

Completness : Not complete if search space is infinite or contains cycles. Complete in finite space

Optimality : Not guarentee finding the optimal sol.

Time Complexity : $O(b^d)$

Space Complexity : $O(b^d)$

▷ Uniform Cost Search : (Cheapest First Search)

↳ Poori working BFS ki hai bas sirf us node ko expand krenge jiski least cost hogi.

↳ Priority queue use hogi

↳ each node is associated with it cumulative cost (cost to reach from root to the current node)

Page No. _____

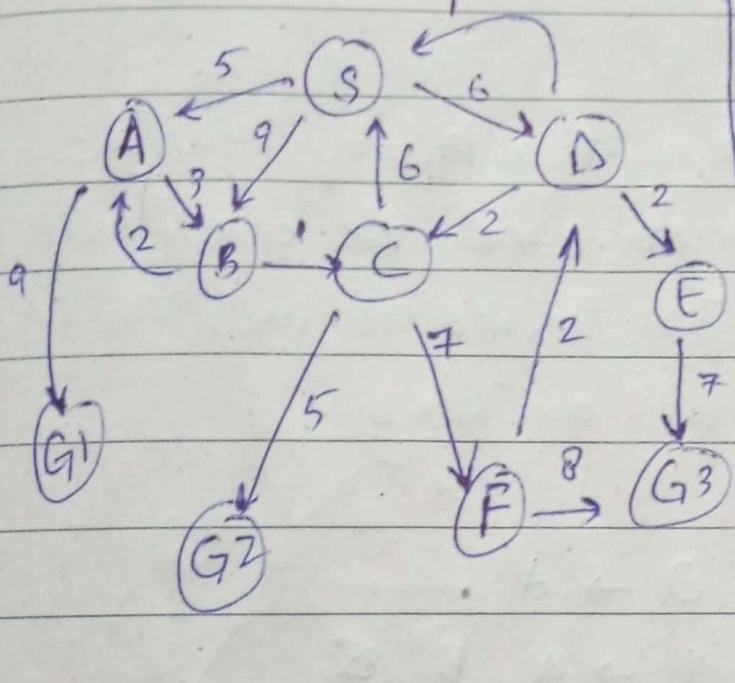
Date _____ Day _____

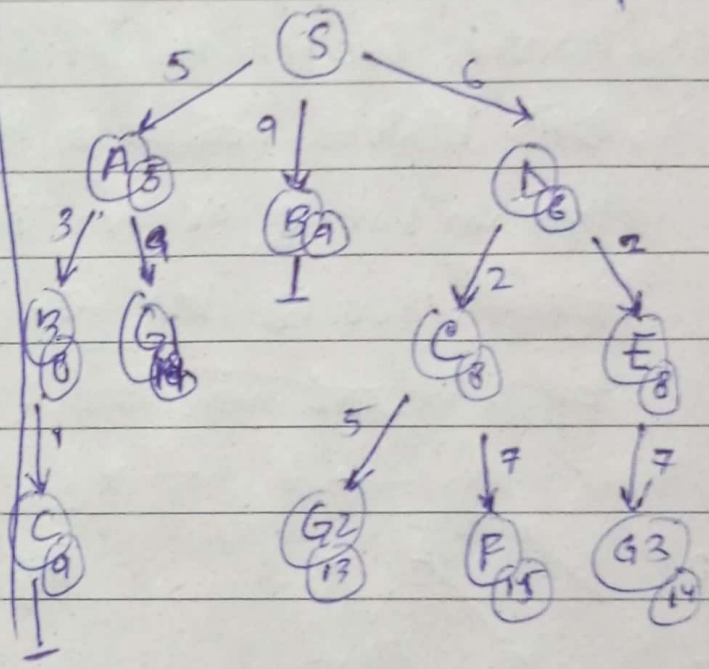
Completeness : Yes

Time Complexity : larger than b^d (space Complexity)

Optimality : Always optimal

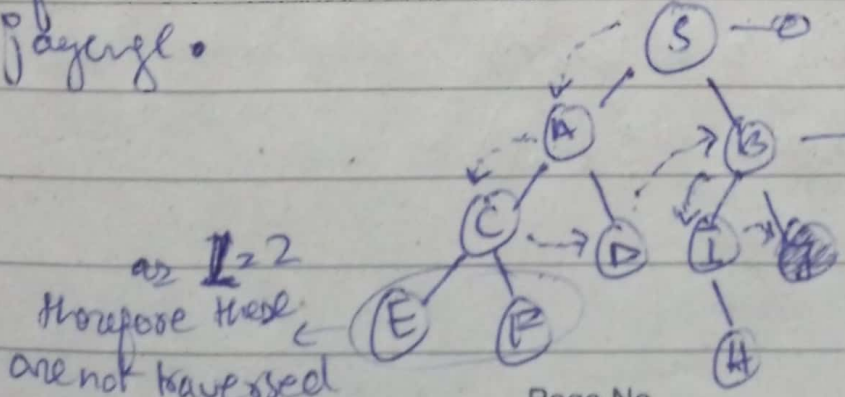
Visited: S, A, D, B, C, E, G2 (goal so stop)





▷ Depth-Limited Search :

↳ DFS ki tarha hota hai bs ismein ek depth limit defined hoti hai. Us depth limit se neeche nhi jayenge like if $l = 2$ then hum root node se sirf 2nd level tak jayenge.

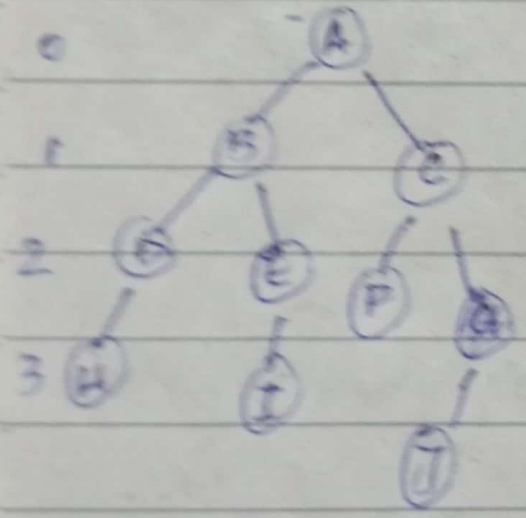


Date _____ Day _____

Completeness : No (if goal is beyond l)
Time : $O(b^l)$ (Space : $O(bl)$)
Optimal : No (if $l < d$)

▷ Iterative Deepning DFS :

↳ DFS ki hi ek hai bas hum multiple iterations perform karenge or har iteration mein ek certain level tak hi traverse karenge. Jese iteration #0 mein level 0 tak, iteration #1 mein level 1 tak and so on.



iter 0 → A.
1 → A, B, C.
2 → A, B, D, E, C, F, G

Completeness : Yes (if b is finite)
Time : $O(b^d)$ Space : $O(bd)$
Optimal : Yes

Date _____ Day _____

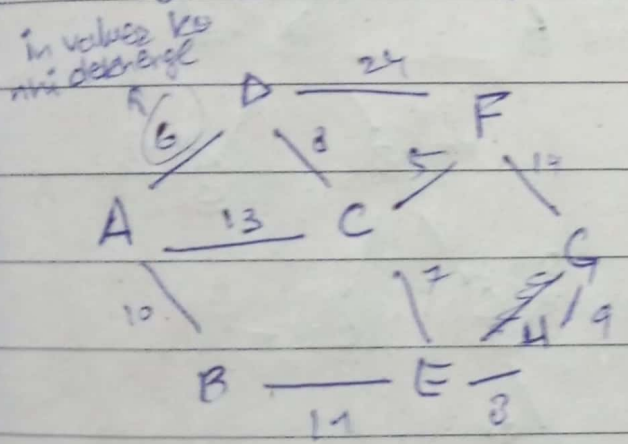
○ INFORMED SEARCH ALGO:

→ A heuristic is a function that estimates how close a state is to a goal.
Ex: Manhatan or Euclidean distance for pathing.

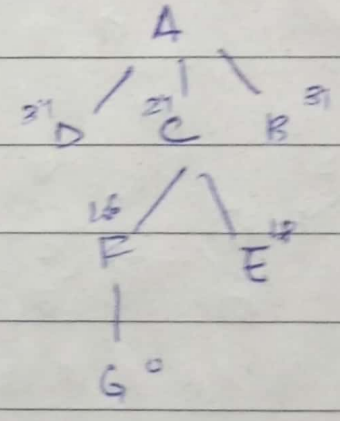
→ Optimality depends upon Heuristic

▷ Best First Search:

G Ismein har node ki ek heuristic value di woi hogi. Ye ek estimated value hogi. Jis node ki sabse kam hogi use hm explore karenge. → ye hs estimated values hai n se goal node tak pahuchne ki



n	h(n)
A	33
B	31
C	24
D	34
E	18
F	16
G	0
H	9



→ Doesnot gives optimal sol.

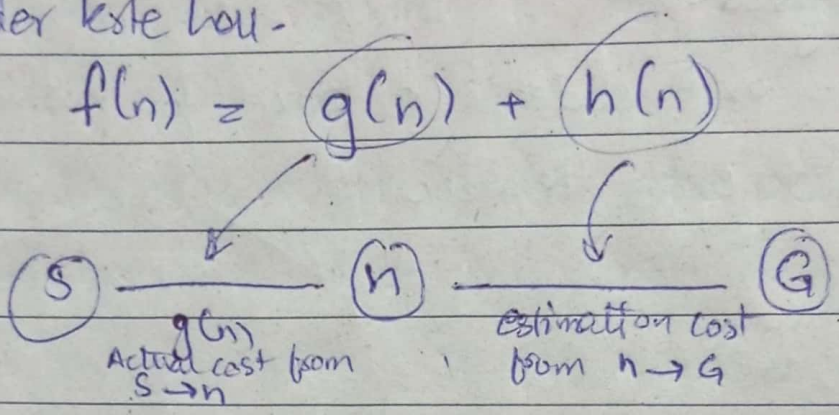
→ Not Complete (May stuck in infinite loop)

Date _____ Day _____

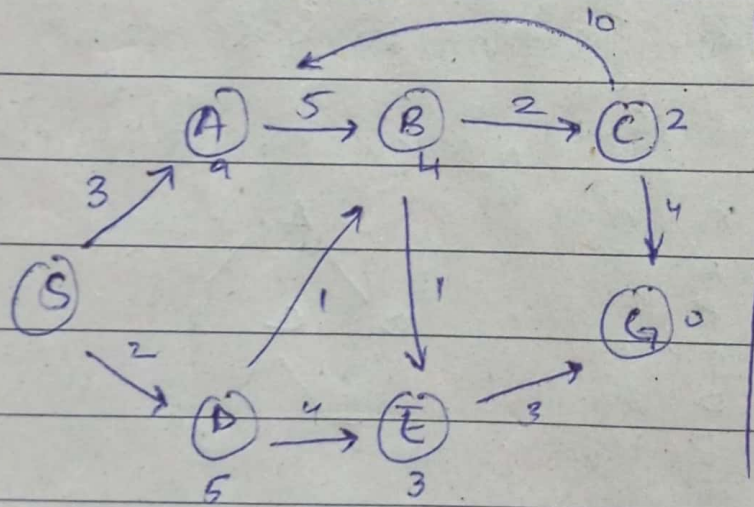
A* Search :

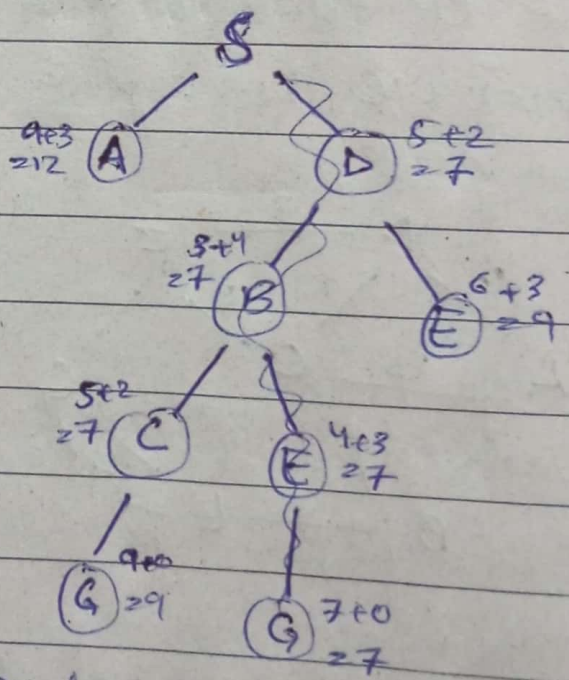
↳ Is sirf h(n) ko nhi dekhte blke actual cost ko bhi consider krte hai.

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



Jiska f(n) kam hoga woh choose krenge





~~FBS~~

Finds optimal solution

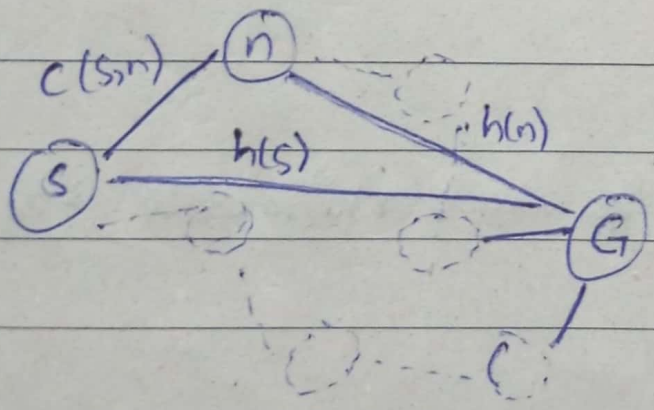
Agr do nodes ki same value aaye to dono ko explore krenge

Date _____ Day _____

- Admissibility For Heuristics
  - → h(n) must be admissible
  - → An admissible heuristic is one that never overestimates the cost to reach the goal.
  - → Admissible heuristic are by nature optimistic becz the think the cost of solving problem is less than it actually is.
  - → Straight line dis is admissible.

- Consistency For heuristic

$$h(s) \leq c(s,n) + h(n)$$



$h(s) \leq c(s,n) + h(n)$

Git is like triangle inequality

- Weighted A* Search:
  - $f(n) = g(n) + W * h(n)$
  - ↳ doesnot gives optimal
  - ↳ but gives quickly

Date _____ Day _____

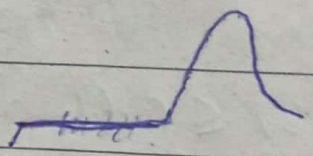
▷ Hill Climbing : (Local Search, Greedy Approach, No backtrac

↳ This algo continuously moves in the direction of incr. elevation / value to find the peak or best sol to the problem.

↳ It terminates when it reaches the peak value where no neighbours has a higher value.

Problems :

① Local Maximum ② Plateau / Flat Maximum



↳ ye algo saari states ka track nhi rakhta sirf current state ko dekhta hai

→ It is not guarenteed to find the optimal sol.
→ It is highly sensitive to the initial state
→ It ~~is~~ doesnot mantain a search history.
→ Req less computation

Date ____________ Day ____________

→ We can perform multiple runs at random initial state to get best possible results.

- Stochastic Hill Climbing: chooses at random from among the uphill moves

▷ Local Beam Search:

↳ Take care of space complexity (constant)

↳ Beam width is given (β)

↳ Saari states ko save karke nhi rakhta, jitni beam value hogi utni hi states save karke rakhega.

▷ Simulated Anealing:

function SIMULATED ANEALING return a sol^n state

input: problem, schedule (a mapping from time to temperature)

local variables: current, next, T (a "temperature" controlling probability of downward steps)

current ← MAKE-NODE(INITIAL-STATE[problem])

for t ← 1 to ∞ do: → T=0 means that I am frozen or ~~frozen means~~ I cannot make any other moves, so return current state

T ← schedule[t]

if T=0 then return current

ΔE → ye hai ki next ki val current se kitni zyada hai.

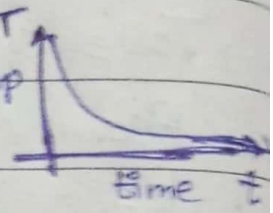
next ← a randomly selected successor of current

ΔE ← VALUE[next] - VALUE[current]

Date

if $\Delta E > 0$ then current ← next
else current ← next only with prob $e^{\Delta E/T}$

→ Is algo mein hum bad moves bhi le sakte hai (like we can also go down the hill)
→ Lekin jab T ki value bohot hight hogi tab hi bad move lene ki probab zyada hogi.
→ T ki value gradually low hoti jayegi.



High T: prob of "locally bad" move is higher
Low T: " " " " " lower

▷ Genetic Algorithm
① Initialize the population (~~solution~~ states) randomly or with potential good solutions
② Compute fitness of each ~~set~~ population ← jiski fitness value achi hogi.
③ Select parents using a selection procedure
④ Create offspring by crossovers & mutation operators
⑤ Compute the fitness of the new offspring
⑥ Select members of population to die using a selection procedure
⑦ Go to step 2 until termination criteria are met.

6+5+3+3+3+2

Date _____ Day _____

→ Hamare paas multiple diff sol" hai or unmein se best sol" extract krna hai

- Types of Crossovers → process of combining two chromosomes (strings) to produce new offspring

① Single Point Crossover:

string 1 → 11011 | 00100110110

string 2 → 11010 | 11000011110

offspring 1 → 11011 | 11000011110

offspring 2 → 11010 | 00100110110

② Two Point crossover

↳ 2 points se break krke combine krenge

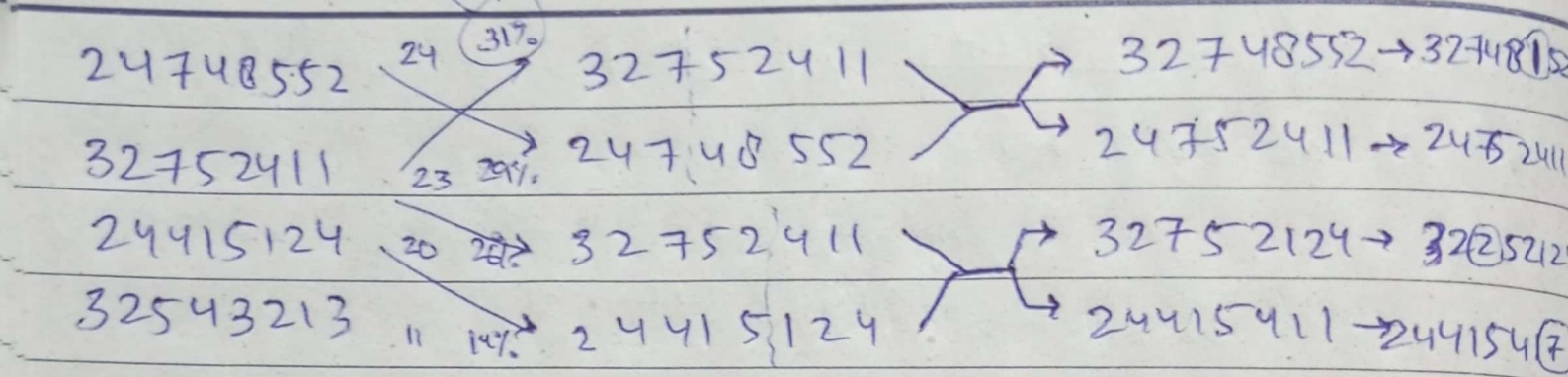
③ Uniform Crossover:

↳ random bits ko include krenge tak ek offspring bnega

jinka zayeda fitness percentage
hogi unko choose karoge $24/78 = 0.307$

$0.307 \times 100 = 30.7\%$ chance of being chosen

Date _____ Day _____



• initial population • fitness function • selection • crossover • mutation

SEND 9 13 18 12

MORE 1 10 15 13

MONEY 10 15 14 13 15