Space for Marks	Question No.	Examiner	Question No.	a	b	c	d	e	f	Total
		Moderator								

3. Problem Solving

Uninformed search methods - Breadth First search (BFS), Depth First Search (DFS), Depth Limited search, Depth First Iterative Deepening (DFID).

Informed search methods - Greedy best first search, A* search, memory bounded heuristic search.

Uniform Search :-

- The term 'uninformed' means they have only information about what is the start state and the end state along with the problem defination.

- The uninformed search techniques also called as 'blind search'.

- These techniques can generated successor states and can distinguish a goal state from a non-goal state.

- All these search techniques are distinguished by the order in which nodes are expanded.

Space for Marks	Question No.		Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

2) a) Depth First search (DFS) :-

- The search tree is expanded depth wise; ie. the deepest node in the current branch of the search tree as expanded. As the leaf node is reached, the search is illustrated

- DFS uses a LIFO fringe, i.e stack.

- The most recently generated node which is on the top in the fringe is chosen first for expansion. As the node is expanded, it is dropped fringe and its successor are added. so when there are no more successors to add to the fringe, the search 'back tracks' to the next deepest node that is still unexplored.

- DFS can be implemented in two way :-
  1) recursive
  2) Non-recursive.

Space for Marks	Question No.		Question No.	a	b	c	d	e	f	Total
			Examiner							
			Moderator							

★ Algorithm :-

1) Recursive implementation:-

1. If node is a goal, return success;

2. for each child c of node

a) if DFS (c) is successful,

i) return success

3. return failure;

★ Non recursive implementation of DFS :-

1. Push the root node on a stack.

2. while (stack is not empty)

a) Pop a node from the stack;

i) if node is a goal node then return success;

ii) Push all children of node onto the stack;

3. return failure.

★ Performance Evaluation -

1) completeness

Space for Marks	Question No.	Examiner	Question No.	a	b	c	d	e	f	Total
		Moderator								

2. optimality - No, as it cannot guarant the shallowest soln.

3. Time complexity -

4. space complexity -

Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total

2. Breadth First search (BFS) :-

- In breadth - First search technique the tree is expanded breadth wise.
- The root node is expanded first, then all the successors of the root node are expanded then their successors and so on.
- All the nodes at a particular depth in the search tree are expanded first and then the search will proced for the Next level node expansion.

⊙ Implementation.

- In BFS we use a FIFO queue for the fringe Because of which the newly inserted nodes in the fringe will automatically be placed after their parents.
- Thus, the childern nodes, which are deeper than their parents go to the back of the queue and old nodes which are shallower, get expanded first.

⊙ Performance Evaluation.

- completeness : provide the shallowest goal node is at some finite depth.
- optimality : as it always finds the shallowest

Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total

- Time complexity: $O(b^D)$, number of node in the fringe.
- Space complexity: $O(b^D)$, total no. of nodes explored.

(c) Algorithm

1. Put the root node on a queue
2. while (queue is not empty)
   a) remove a node from the queue
      i) if (node is a goal node) return success
      ii) put all children of node onto the queue;
3. return failure;

Space for Marks	Question No.	Question No.	a	b	c	d	e	f	Total
		Examiner							
		Moderator							

3. Depth Limited search (DLS) :-

- In order to avoid the infinite loop condition arising in DFS, in depth limited search technique, depth First search is carried out with a predetermined depth limit.
- The nodes with the specified depth limit are treated as if they don't have any successors. The depth limit solves the infinite-path problem.
- DFS can be viewed as a special case of DLS with depth limit equal to the depth of the tree.

@ Implementation:-

- As in case of DFS in DLS we can use the sam fringe implementation as queue.

- Additionally the level of each node needs to be calculated to check whether it is within the specified depth limit.

Depth-limited search can be terminate with two conditions:

1. If the soln is found
2. If there is no soln within given Depth limit.

Space for Marks	Question No.	Examiner	Question No.	a	b	c	d	e	f	Total
		Moderator								

Algorithm :-

1. Determine the start node and the search depth.

2. Check if the current node is the goal node
   1. If not : Do nothing
   2. If yes : return.

3. Check if the current node is within the specified search depth
   1. If not : Do nothing
   2. If yes : Expand the node & save all of its successors in a stack.

4. Call DLS recursively for all nodes of the stack & go back to step 2.

Performance Evaluation.

1. Completeness - its incomplete if ~~Provided the~~ shallowest goal is beyond the depth limit. ~~node is at some finite depth.~~
2. Optimality - As it always finds the shallowest soln.
3. Time complexity - $O(b^d)$, no. of node in the fringe.
4. Space complexity - $O(b^d)$, total no. of nodes expired.

Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

2. Optimality - Non optimal, as the depth chosen can be greater than d.

3. Time complexity - same as DFS, $O(b^l)$, where l is the specified depth limit.

4. Space complexity - same as DFS, $O(b^l)$, where l is the specified depth limit.

4. Depth First Iterative Deepening search (DFI

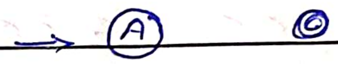
- Depth First Iterative Deepening search is a combination of BFS & DFS. In DFIDS happens depth wise but at a time the depth limit will be incremented by one. Hence iteratively it deepens down in the search tree.

- It eventually turns out to be the BFS as it explore a complete layer of new nodes at each iteration before going on to the next layer.

- It does this by gradually increasing the depth limit - first 0, then 1, then 2 and so on - until a goal is found; and thus guarantees the optimal soln. Iterative deepening combines the benefits of depth-first & de breadth-first search.

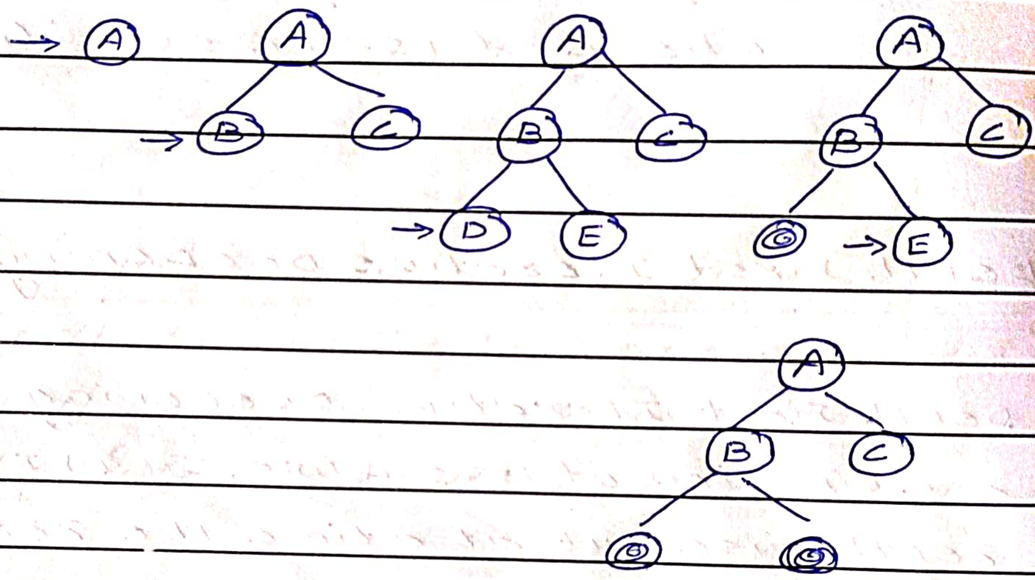
Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

Limit = 0

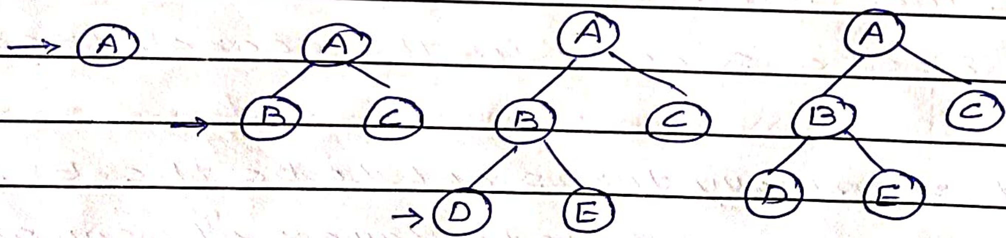


Limit = 1

Limit = 2



Limit = 3



Space for Marks	Question No.		Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

Implementation:

It has exactly the same implementation as that of DLS. Additionally, iterations are required to increment the depth limit by one in every recursive call of DLS.

Algorithm

1. Initial depth limit to zero.
2. Repeat until the goal node is found.
   1) call depth limited search with new depth limit.
   2) Increment depth limit to next level

Performance Evaluation:-

1. completeness - DFID is complete when the branching factor b is finite.
2. Optimality - It is optimal when the path cost is a non-decreasing function of the depth of the node.
3. Time Complexity - is $O(b^d)$
4. space complexity - memory requirement of DFID are modest i.e $O(b^d)$.

Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

** Informed Search methods :-

9. greedy best first search :-

- A greedy algo. is an algorithm that follows the heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum.

- when BFS uses a heuristic that leads to goal node, so that nodes which seems to be more promising are expanded first. This particular type of search is called greedy best-first search.

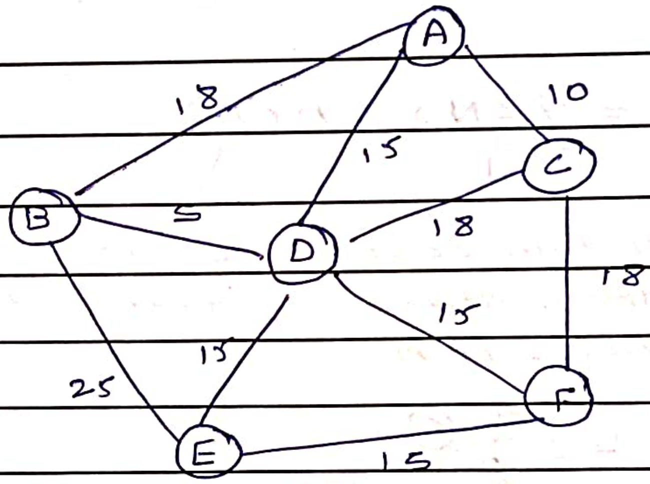
- greedy best first search may not always produce an optimal soln, but the soln will be locally optimal, as it will be generated in comparatively less amount of time. In mathematical optimization, greedy algo. solve combinatoric problems.

- example:

consider the traveling salesman problem which is of a high computational complexity works well with greedy strategy as follows. The values written on the link are the straight line distance from the node. Aim is

...ce ...or ...rks	Question No.		Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

∴ to visit all the cities A throught F with the shortest Distance travelled.



it will always make a local optimal choice Hence it will select node C first as it found to be the on with less Distance from the next non-visited node from node A, and then the path generated will be the optimal path and optimal Distance the salesman needs to travel.

A → B → D → E → F → C where the cost to be 18 + 5 + 15 + 15 + 18 = 68.

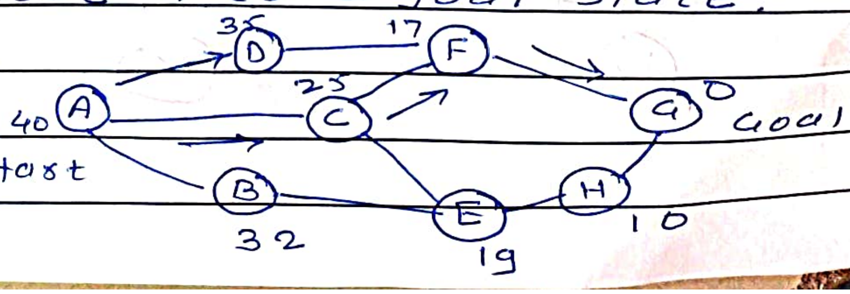
$$f(n) = h(n)$$

∴ h(n) = estimated cost from the state at node n to a goal state.

A → C → F → G

40 + 23 + 17 + 0

=



Space for Marks	Question No.	Examiner	Question No.	a	b	c	d	e	f	Total
		Moderator								

20 A* search :- admissible

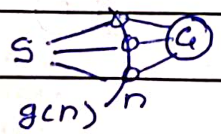
Allready knowledge heuristic

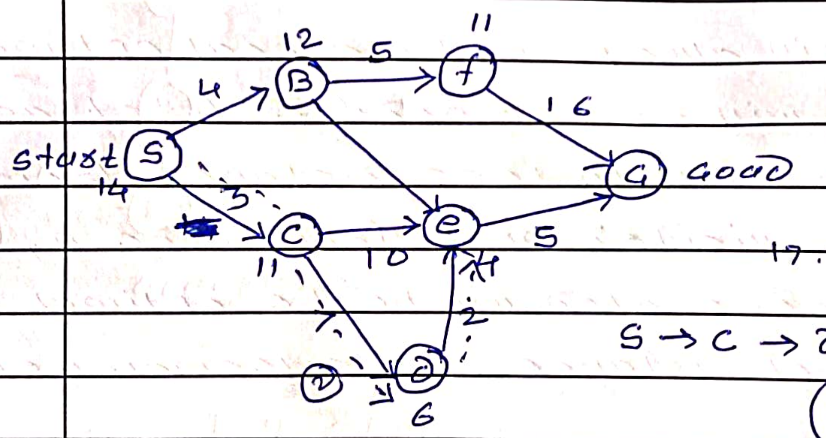
- Informed searching technique.

- $f(n) = g(N) + h(N)$.

Actual cost from start node to n.

estimation cost from n to goal node.





S → C → D → e → G

17

$f(S) = 0 + 14 = 14$

S → B

4 + 12

16

S → C

3 + 11

14

SCD → e

3 + 7 + 2 + 4

10 + 2 + 4

16

~~SC → e~~

3 + 10 + 4

17

~~SC → D~~

3 + 7 + 6

16 //

SCDe → G

~~3 + 7 + 2 + 5 + 0~~

10 + 7

17 //

Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

- A* algorithm is also called as OR graph / tree search algo.

- In A* search, the value of a node n, represented as f(n) is a combination of g(n), which is the cost of cheapest path to reach to the node from the root node, and h(n), which is the cost of cheapest path to reach from the node to the goal node.

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

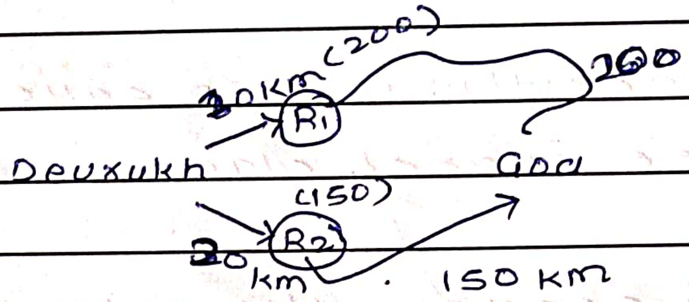
* **Memory Bounded Heuristic search.**

- It is a technique designed to solve a problem quickly.

- Uninformed search – bliend
  Informed search.

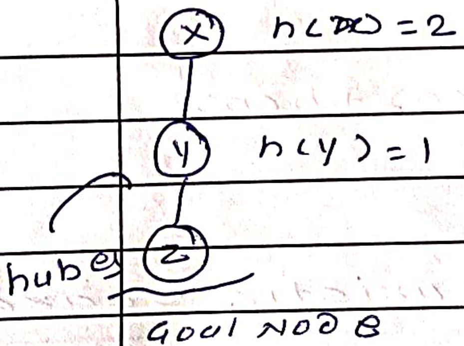
- It is a funⁿ H(n) that gives as estimation on the cost of getting from node 'n' to the goal state.

Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total



$\therefore R_1 = 10 + 200 = 210$

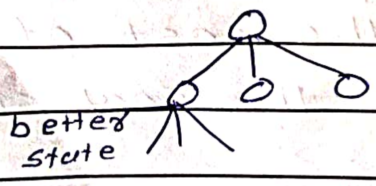
$R_2 = 20 + 150 = 170/-$



Simple hill climbing

- Local search (Globle Domain knowledge Nahi)
- greedy approach - best move
- No backtrack -

- Evaluate initial state



Not back tracking

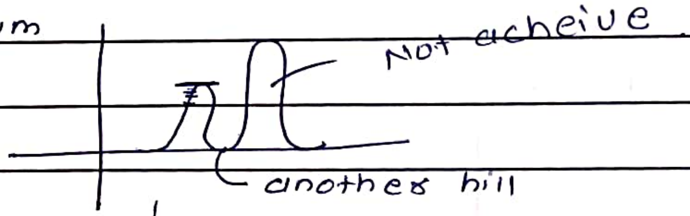
Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

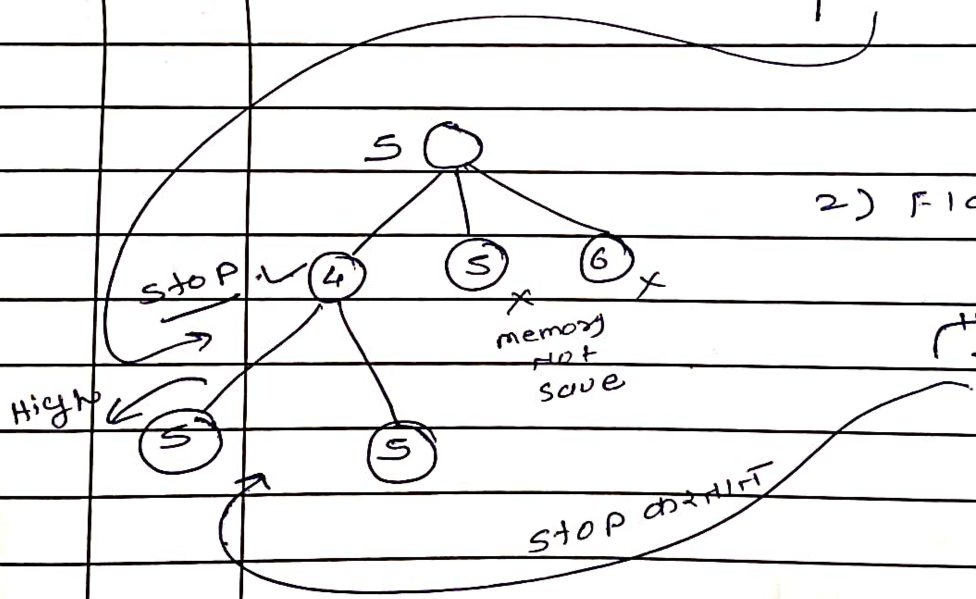
if better than current state then it is new current state.

– minimum value selet.

9) Local problems in hill climi

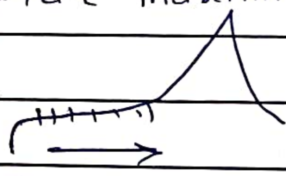
1) Local maximum





Plateau

2) Flat maximiu

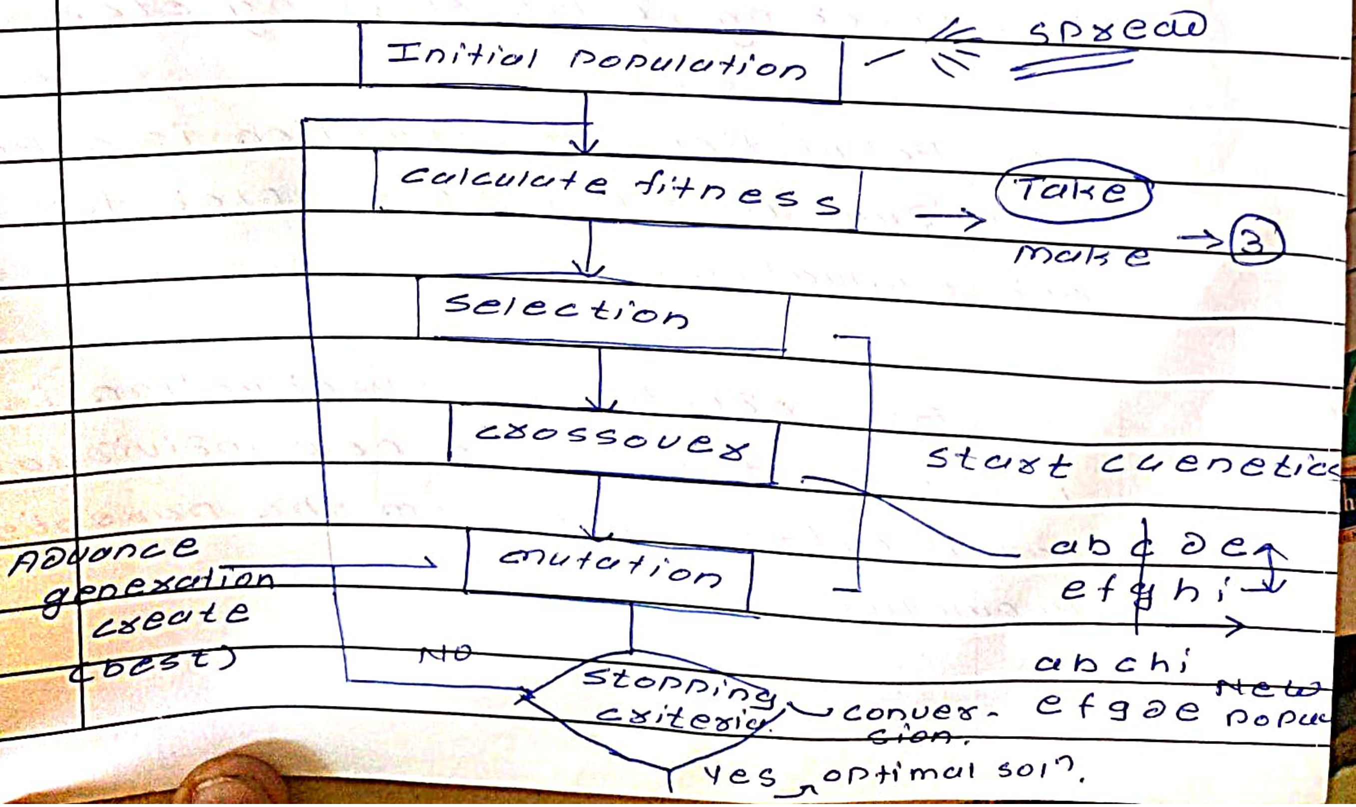


3) Ridge

Question No.		Question No.	a	b	c	d	e	f	Total
	Examiner								
	Moderator								

* Genetic Algorithm :- (John Holland).

- GA algorithm are adaptive heuristic search algo. based on the evolutionary ideas of natural selection and genetics.
- GAs are by no. means random.
- GA are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype or the genome) of candidate soln (called individuals, creatures, or phenotypes) to an optimization problem evolves towards better soln. The soln are represented in binary as strings of 0s & 1s, but other encodings are also possible.



Space for Marks	Question No.	Examiner	Question No.	a	b	c	d	e	f	Total
		Moderator								

★ Terminologies of GA:

1. Gene : Gene is the smallest unit in genetic algo.

2. chromosome : chromosome is a series of genes that represent the components of one possible solⁿ to the problem.

3. Encoding : Encoding of chromosomes is one of the problems to start solving problem with GA. Encoding depends on the type of problem.
These are various types of encoding techniques like binary encoding, permutation encoding, value encoding etc.

4. Population - A population is a pool of individual that will be sampled for selecti and evaluation.

5. Reproduction - Reproduction is the process of creating new individuals called off-springs from the parents population.

Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
	Examiner								
	Moderator								

★ Genetic operation -

1. selection -
chromosomes are selected from the population to be the parents for the cross over. The problem is how to select chromosomes.
E.g. roulette wheel selection, rank selection, steady state selection.

2. crossover - crossover involves the exchange of gene info^n bet^n two selected chromosomes.

3. mutation - mutation is another refinement step that randomly change the value of a gene from its current setting to a completely different one.
Types of mutation techniques like bit inversion, order changing, value encoding.

Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

* Example:

1. parent chromosomes.

A 1 5 3 2 6 4 7 9 8

B 8 5 6 7 2 3 1 4 9

child chromosome after arithmetic crossover. + bits of both chromosomes

C 9 0 9 9 8 7 8 3 7.

* After appling value encoding mutation.
∴ i.e + or - a small value to selected values e.g. 1 to 3rd & 4th bit.

2. child chromosome

C 9 0 8 8 8 7 8 3 7.

Question No.	Examiner	Question No.	a	b	c	d	e	f	Total
	Examiner								
	Moderator								

** 3.3 Adversarial Search :- 

- Adversarial search problem is having competitive activity which involves 'n' players and it is played according to certain set of protocols

- Game is called adversarial because there are agents with conflicting goals and the surrounding environment in a game is competitive as there are 'n' players or agents participating.

- As the actions of every agents are unpredictable there are many possible moves.

- In order to play a game successfully every agent in environment has to first analyze the action of other agents and how other agents are contributing in its own wining or lossing. After performing this analysis agent executes.

* Environment Types.
  1) competitive environment
  2) cooperative environment.

Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

1) competitive environment :

– In this type of environment every agent makes an effort to win the game by defeating or by creating superiority over other agents who are also trying to win the game.

– chess is an e.g of a competitive environment.

2) cooperative environment.

– In this type of environment all the agents jointly perform activities in order to achieve some goal.

– car driving agent is an e.g. of a cooperative environment.

//

– It is a game-playing technique where agents are surrounded by a competitive environment. A conflicting goal is given to the agents.

– Game-playing means discussing those games where human intelligence & logic factor is used. Tic-Tac-Toe, chess.

- This search is based on the concept of 'Game Theory'.

- To complete game, one has to win the game & other looses automatically.

my interest		your interest
I win		you loose.

* Types of algorithms in Adversarial search.

- In a normal search, we follow a sequence of action to finish the game optimally. But in an adversarial search the result depends on the players which will decide the result of the game.

- It is also obvious that the soln for the goal state will be an optimal soln because the player will try to win the game with shortest path and under limited time.

- There are following types of adversarial search:
  - min max Algo
  - Alpha-beta Algo.

Space for Marks	Question No.	Examiner	Question No.	a	b	c	d	e	f	Total
		Moderator								

9. *** Game Playing :-

* Types of Gaming :- (2)

1) Deterministic :-

- It is a fully observable environment, when there are two agents playing the game alternatively and the final results of the game are equal and opposite then the game is called deterministic.

- example.

Tic-Tac-Toe where two players play a game alternatively and when one player wins a game then other player losses game.

2) Probabilistic

- Probabilistic is also called as non-deterministic type. It is opposite of deterministic games, where you can have multiple players and you cannot determine the next action of the player.

- Example.

You can take example of card game

Question No.	Question No.	a	b	c	d	e	f	Total
	Examiner							
	Moderator							

9)
- Game playing is also known as adversarial search

- Games are adversarial search problems.

- Most of the games are deterministic

- Example: Two main varieties of problems faced in artificial intelligence game.
  1) Toy Problems
  2) Real Problems.

Toy problems	Real Problems
1) 8-queen	1) Traveling salesperson.
2) vacuum world	2) Robot Navigation
3) missionaries & cannibals.	3) Assembly sequencing.

Space for Marks	Question No.	Examiner	Question No.	a	b	c	d	e	f	Total
		Moderator								

(2) * * Min max Algorithm :-

- minimax algo evaluates decision based on the present status of the game. This algo. needs deterministic environment with perfect infoⁿ.

- min max algo. Directly implements the defining eqⁿ. Every time based on the successor state minimax value is calculated with the help of simple recursive computation.

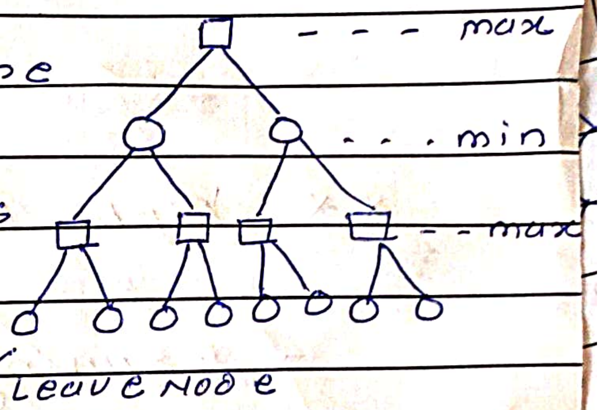
- In case of minimax algo. the selected action with highest minimax value should be equal to the best possible payoff against best play.

* - consider the following two player game tree in which the static scores are given from the first player point of view.

- Apply the min max search algo. & compute the value of the root of the tree.

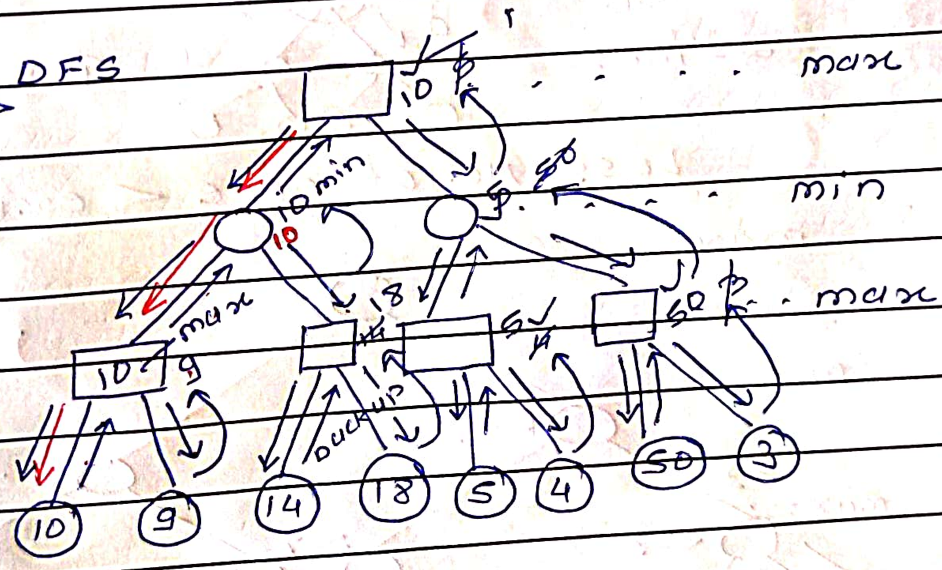
- Also find the most convenient path for max Node.

Question No.		Question No.	a	b	c	d	e	f	Total
	Examiner								
	Moderator								



- Generate the whole game tree to leaves.
- Apply utility fun^n to leaves
- Use DFS for expanding the tree
- Back-up values from leaves toward the root:-
  1) a max node compute the maximum value from its child value.
  2) a min node compute the mimimum value from its child node
- When value reaches the root: optimal move is determined.

* Use DFS



∴ Root Node 10

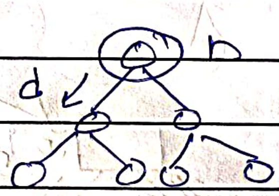
Space for Marks	Question No.	Examiner / Moderator	Question No.	a	b	c	d	e	f	Total
		Examiner								
		Moderator								

~~[illegible]~~ (~~Assume~~)

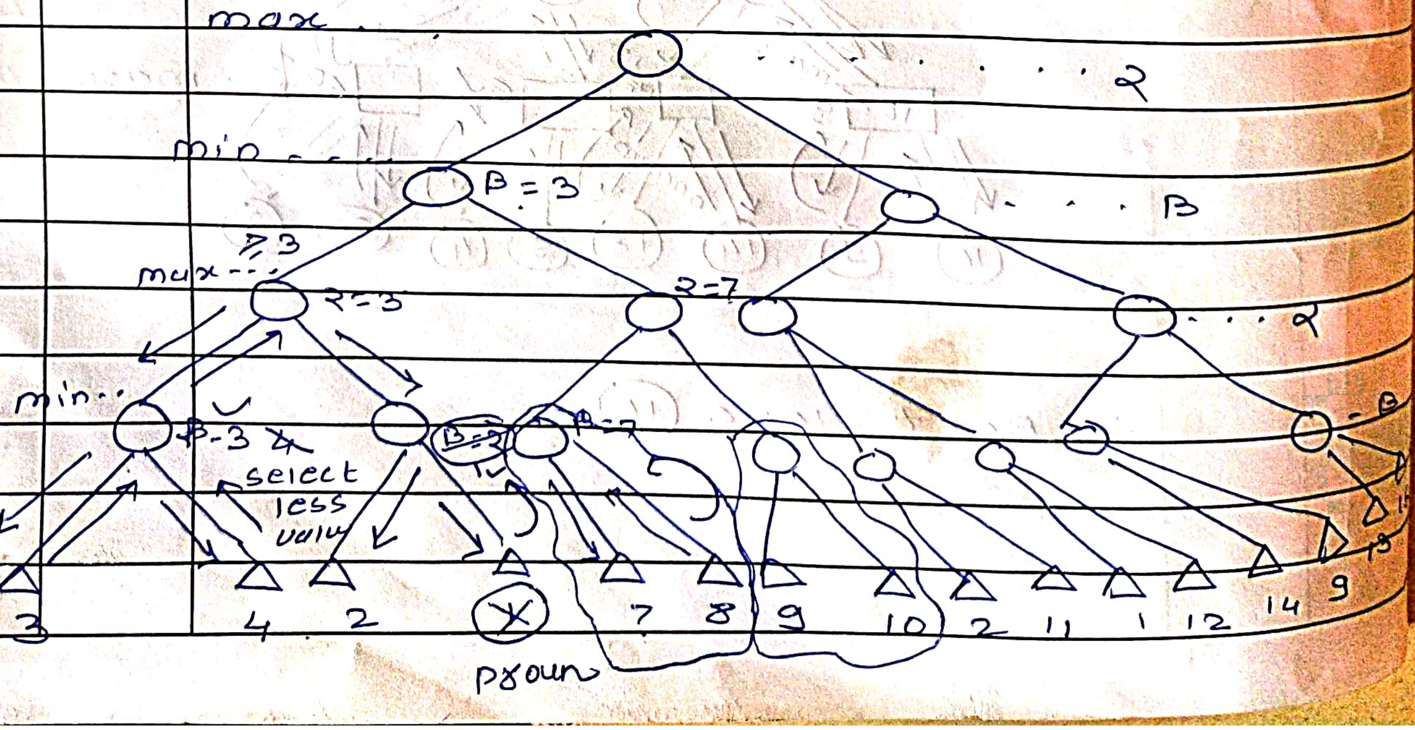
** Alpha Beta Pruning :- DFS.

- Pruning means cutting off. In game search it resembles to clipping a branch in the search tree, probably which is not so fruitful.
- α-β pruning is an extension to minimax algo where, decision making process need not consider each & every node of the game tree
- cut off search by exploring less no. of nodes

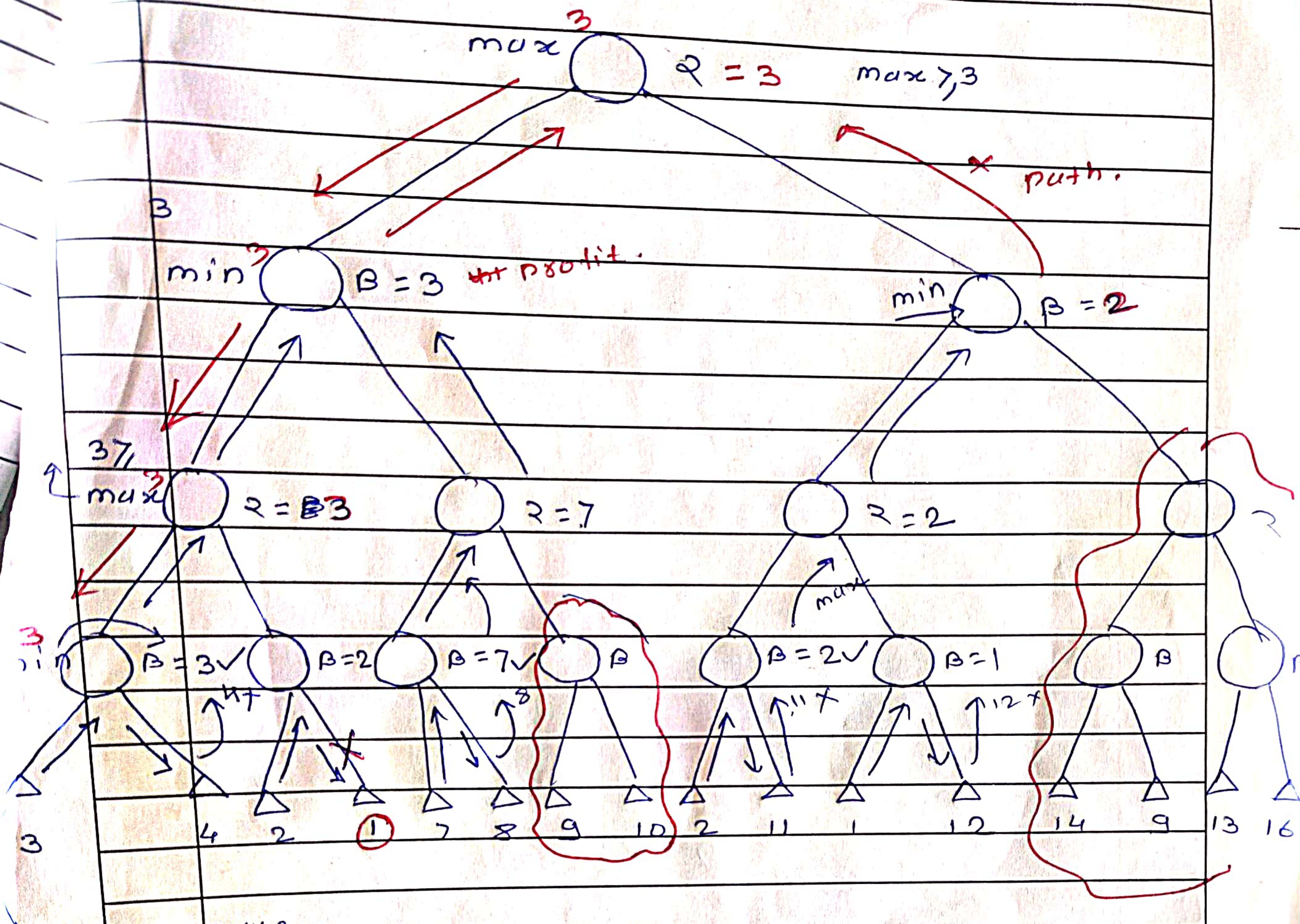
performance matrix $O(b^{d})$.

α - max  β - min.

Less No. of node < explore કરશે.



DFS USED 4)



Terminal value.