LMT

(Last Moment tuitions) ← youtube channel which makes complex things easy

NOTES

(THIS NOTES ARE CLEARLY MADE FOR THOSE WHO STUDIES AT AT LAST MOMENT BECAUSE HAME PATA H LAST MOMENT KI Situation kya hoti hai:)

BEST OF LUCK FOR EXAM.

Before seeing notes we suggest you to see video LAST MOMENT TUITIONS PE JAKE USME CONCEPT Bahut ache se padhaya hai uske baad notes will be very easy =)

PAGE:

PLEASE YEAH
NOTES AUR VIDEO
SHARE KARO UNKO
JO APKE JITNE SMART
NAHI TAKI VOH
BHI ACHE MARKS
LAYE APKI TARAH

LMT

(LAST MOMENT TUITIONS)

EXPERIMENT : No. BFS

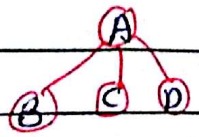
PAGE No.
DATE

Uninformed Search Strategies

Uninformed Search Strategies have no additional information about states beyond that provided in the problem definition.

There are five uninformed strategies
1) - Breadth first search

- BFS is a simple strategy the root node is expanded first then all their successors are expanded next, then their successors and so on.
- In general all the nodes are expanded at a given depth in search tree before expanding nu node at next level
- Uses two list
  Open : list of genrated nodes but not yet expanded

(Suppose 

A ko expand kiya toh B, C, D generate hua par voh expanded nahi hai that comes in open)

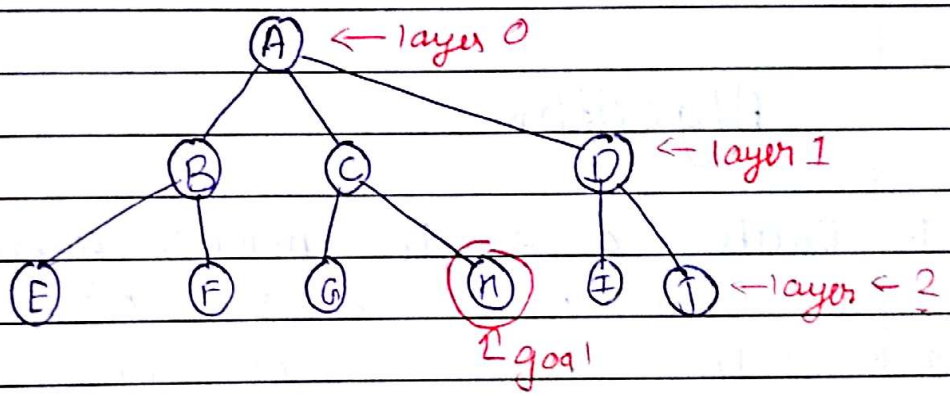
EXPERIMENT : No.

PAGE No.
DATE

Close : (Nodes which are visited and explored or expended)
visit bhi ho chuke hai aur explored bhi

- BFS is implement in FIFO manner. (Open list in FIFO) . i.e nodes that are visited first are expanded first

Example :



Open [A]
X = A
close = [A]

Open (B, C, D)
X = B
close [A, B]

OPEN (E, D, F, F)
X = C
close (A, B, C]

Open (D, E, F, G, H)
X = D
close (A, B, C, D)

Open (E, F, G, H, I, J)
X = E
close (A, B, C, D, E)

Open (F, G, H, I, J)
X = F
Open (G, H, I, J)
X = G
Open (H, I, J)
X = H [Goal success]
Open (I, J)
X = I
Open (J)
X = J

Close (A, B, C, D, E, F)
Close (A, B, C, D, E, F, G)
Close (A, B, C, D, E, F, G, H)
Close (A, B, C, D, E, F, G, H, I)
Close (A, B, C, D, E, F, G, H, I, J)

Algorithm

1. Create a single number queue comprimising (containing) of root node
2. If the first number of the queue is goal node then go to Step 5
3. If the first member of the queue is not the goal node then
   i) Remove it from open node
   ii) Add it to visited list
   iii) Consider its children if any and put them in queue from Backside
4. If the queue is not empty go to Step 2
5. If the queue is empty go to Step 6
   Print success and Stop
6. Print failure and Stop.

Sundaram

Teacher's Sign.:

EXPERIMENT : No.

PAGE No. DATE

Explanation :- 1) Ek queue lo jisme root node ko dala.

2) check karo Pehla number jo hamne queue me dala jo ki root node tha voh goal hai kya. agar hai toh step 5 Pe ja

3) Agar nahi hai toh
- Usko open list se nikal
- visited list me add kar
- Uske koi children hai ki nahi check kar aur agar hai toh unko queue me add kar piche se.

4) queue khali hai kya Toh step 6 Pe ja. Nahi hai toh step 2 Pe ja

5) mil gaya goal success Print karo aur Stop karo

6) Goal nahi mila failure stop karo Abhi.

Advantages :-
1) completeness → Yes
2) Optimum → gives shallowest goal

Disadvantages:
1) Space complexity : $O(b^{d+1})$ Big 'O' Notation
where b- branching factor
d - is depth of goal.
2) Time complexity : $O(b^{d+1})$

Sundaram

Teacher's Sign.:

DFS
(Depth first search)

In this the current node is expanded till depth ~~then go to~~ till where there is no successor or children. Then goto the next one.

The DFS is implemented using LIFO Strategy.

There are 2 list

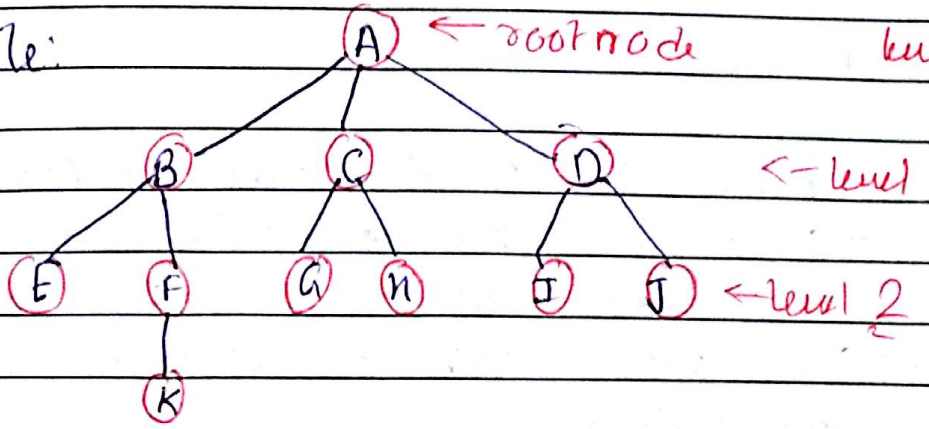
Open : Nodes which are visited but not explored or expanded

close : Nodes which are visited and explored

EXPERIMENT : No.

PAGE No.
DATE

Example:



X = A	Close ( )
Open (B, C, D)	close (A)
X = B	
Open (E, F, C, D)	close (B) (A, B)
X = E	
Open (F, C, D)	close (E) (A, B, E)
X = F	
Open (K, C, D)	Close (A, B, E, F)
X = K goal node Success	

Algorithm

1) • Create a single number queue comprimising of root node

2) • If the first number of the queue is Goal then Goto Step5

3) • If the first number of the queue is not goal node then
   i) Remove it from open fringe node (list)
   ii) Add to the list of visited node
   iii) Consider the children or successor if any which are not yet explored and add them to queue from front side (LIFO)

4) • If the queue is emply then goto step6
   If the queue is not emply goto Step2

5) Print SUCCESS & STOP

6) Print failure & stop

Advantages

1) Space : completity : O(bm)
b - braneing factor
m - depth of solution

2) Time complexity: $O(b^m)$ → Exponential

Disadvantage

1) Completeness - NO
2) Optimum - NO

Depth limit Search (DLS)

It is a strategy where search puts limits to the maximum depth so that infinite loop can be avoided
- That is node at "depth l" will be consider as they dont have any successors or children.
- DFS can be viewed as a special case of DLS where depth limit is ∞.

Note there can be two failure

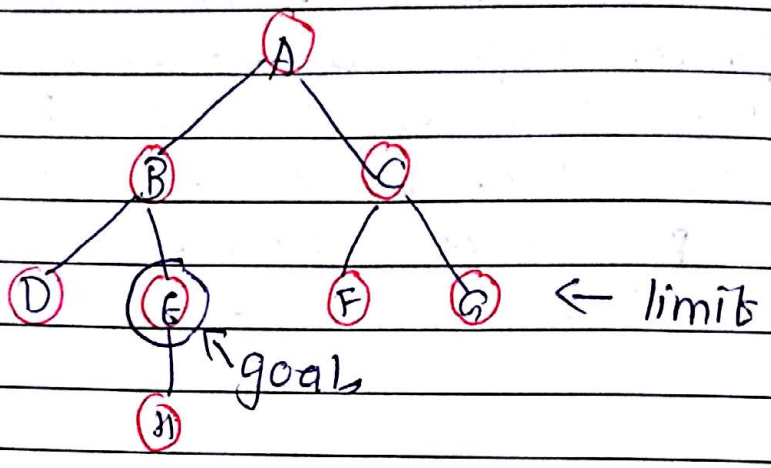
① No solution found
② No solution found with in the limit.

Advantage: Remove infite loops

Disadvantage: Not complete because if depth where there is goal is greater then limit then we will not get the goal or solution
Optimal: No (if d < e)

Time complexity = $O(b^l)$
Space complexity = $O(bl)$

Example



X = A

Open (B, C) close (A)

X = B

Open (D, E, B, C) close (A, B)

X = D

Open (E, B, C) close (A, B, D)

X = E ⇐ goal node reach

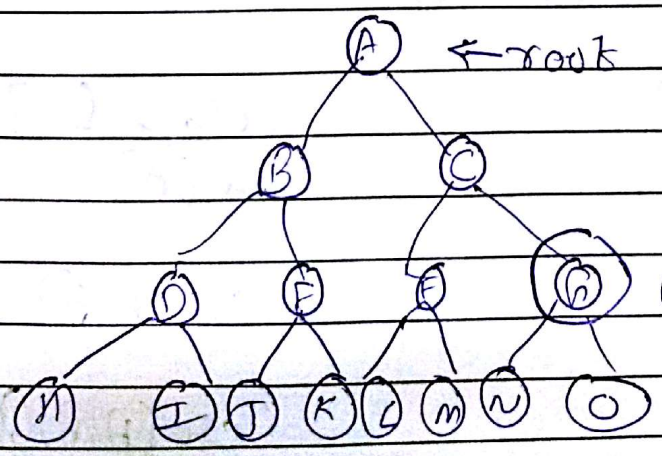
Thus we dont have to go to level 3 even if the goal was or will be in the level 3 we are not allowed.

EXPERIMENT : No. Page No: 11

PAGE No. DATE

(IDS) Iterative Deepning Dept first Search

- Improved DLS
- IDFS is a genural Strategy which uses combinateon of DFS and BFS to remove incompleteness of DLS.
  - or to find Best depth limit
- It is done by gradually increasing limit by 0 then 1 then 2 and so and until we get best depth limit i.e goal
- most popular uninformed search.



Teacher's Sign.:

EXPERIMENT : No. Page No: 12

PAGE No.
DATE

iteration 1
Dept 0

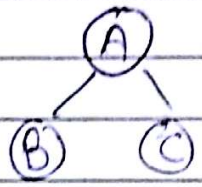
(A) ← 0

Open(A)
x = A
Open[ ]

close[ ]
close[A]

iteration 2

Dept 1



Open[A]
x = A
Open(B, C)
x = B
open[S)
x = C

close[ ]
close[A]

close(A, B]

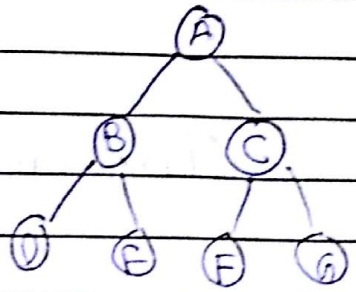
close[A, B, C)

Sundaram

Teacher's Sign.:

EXPERIMENT : No. Page No: 13

PAGE No. DATE



iteration 3
Dept 2

Open [A]
X = A
close [A]
Open [B, C]
X = B
close (A, B)
Open (D, E, C)
X = D
close (A, B, D)
Open (E, C)
X = E
close (A, B, D, E)
Open (C)
X = C
close (A, B, D, E, C)
Open (F, G)
X = F
close (A, B, D, E, C, F)
Open (G)
X = G ← Goal found

Algorithm Same as DFS

Advantages: Combines the benefit of both bfs and DFS
Completeness and optimum → As BFS
Less Space complexity → As DFS

Disadvantages: Wasteful because nodes are generated multiple times.

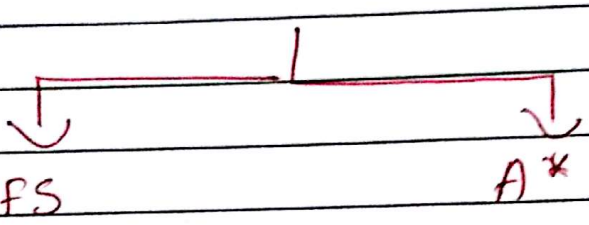
Sundaram

Teacher's Sign.:

EXPERIMENT : No. Page No:14

PAGE No.
DATE

Informed Searches



Informed search : Voh search hai jisme hame goal bhi pata hota hai aur rasta bhi

GBFS : Greedy Best first Search

yeah banda lalchi hota hai jo sasta mila leliya kam ka hai bhi ki nahi voh yeh nahi dekhta

formala

$f(n) = h(n)$

(yeah hame calculate karna padta hai h se yani intermidiate node se goal tak ki jo distance hoti hai voh hota hai $h(n)$ j

Sundaram

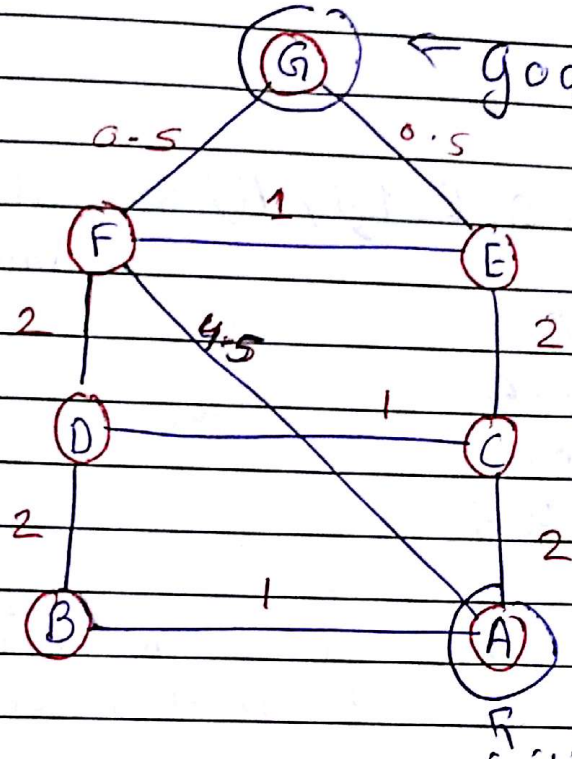
Teacher's Sign.:

EXPERIMENT : No.

Page No: 15

PAGE No.
DATE

Example



Open(A)

X = A → close (A)

open(B, F, C)

ab hum function nikalenga

F(B) = B se goal taki value i.e(2+2+0.5)
= 4.5

F(F) = F se goal node tak ki value
0.5

F(C) = C se goal node tak ki value
2.5

Cab app C se D and F se bhi ja sakte hai par hamesha optimal value (less ki) yani best value

Sundaram

Teacher's Sign.:

EXPERIMENT : No. Page No: 16

PAGE No.
DATE

X = F

open (G, E, D, A) ← yeah wapas A pe bhi jayenga.

f(G) = 0
f(E) = 0.5
f(D) = 2.5
f(A) = 4.5

Thus G is minimum and hume we got goal node But voh optimal nahi hai hame path mila

A - F - G = 5

Par 4.5 ie A - C - E - G optimal tha.

yeah iska disadvantage hai.

Iska solution hai A* Problem, or A* Algorithm.

Sundaram

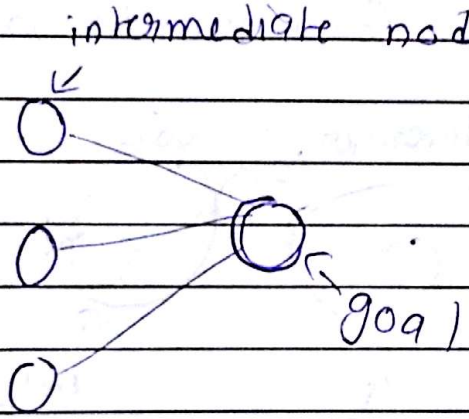
Teacher's Sign.:

EXPERIMENT : No.

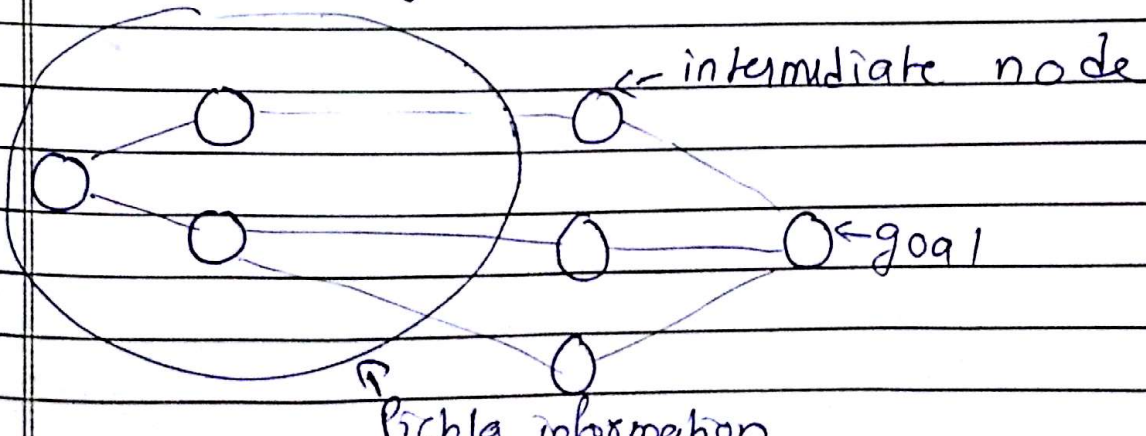
PAGE No.
DATE

(A* Algorithm with solved example)

GBFS me kya hota tha voh na goal ki taraf Bhagta hai



A* Pichla data dekhta goal ke Pass jata hai



Sundaram

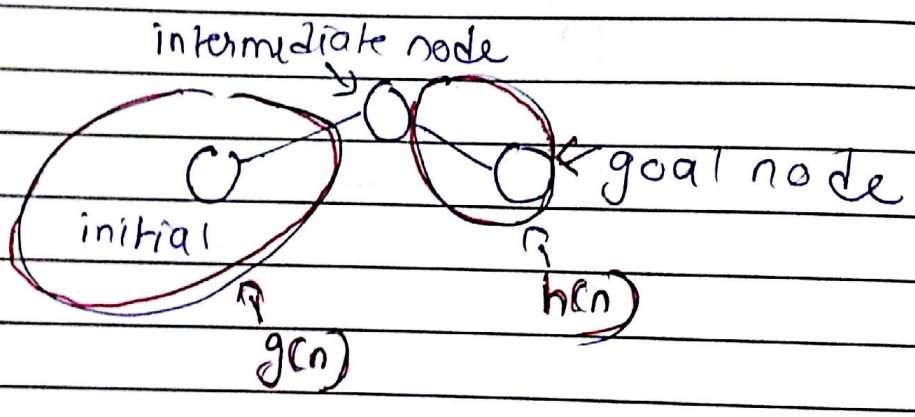
Teacher's Sign.:

Toh iska formula hota hai

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

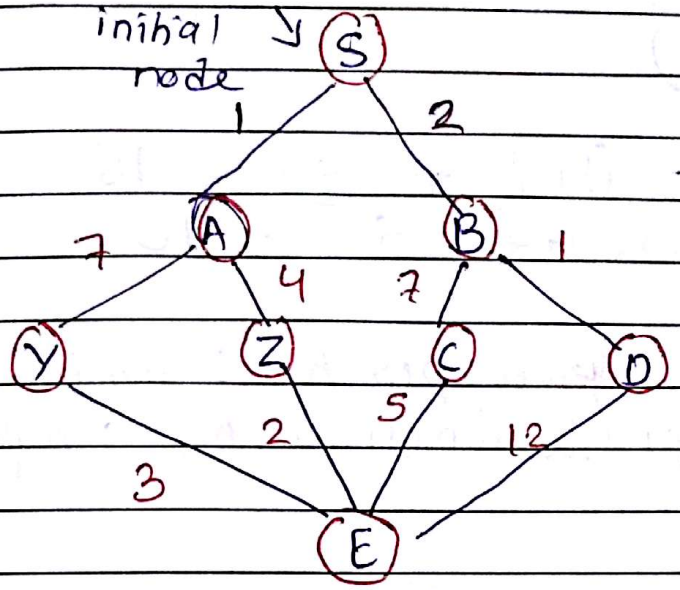
$g(n)$ - Pichla information

$h(n)$ - estimated goal tak Pahuchne ki value



Yeah hame optimal solution deta hai.

Example :-



S = 10
A = 5
B = 6
C = 4
D = 15
Z = 5
Y = 8

OPen (S)
X = S

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

↖ yeah count hota nahi hai par nikalke rakhna padhta hai

OPen (A, B)
$f(S-A) = g(n) + h(n) = 1 + 5 = 6$
$S(S-B) = g(n) + h(n) = 2 + 6 = 8$

ab dono me minimum konsa hai = 6
Toh Ab hum A ko expand karunge

EXPERIMENT : No.

PAGE No.
DATE

X = A

Open (Y, Z)

f(S-A-Y) = (1+7) + (8) = 16
f(S-A-Z) = (1+4) + 5 = 10

ab hum yeah upar ke 2 function aur pichle function ko compare karenge

Toh hum (16, 10, 8)
(yeah pichla jo function tha usse pehla jo explore nahi hua)

ab tino me minimum kya hai

& jo f(S-B) ka function hai.

Toh ab B ko explore ya expand karenge.

Teacher's Sign.:

X = B

Open(C, C, D)

f(S-B-C) = (2+7) + 4 = 13
f(S-B-D) = (2+1) + 15 = 18

Hirse compare karunge sab function ko

min(f(S-B-C), f(S-B-D), f(S-A-Y), f(S-A-Z))

min(13, 18, 16, 10)

Thus we will explore Z
ab hum Z ko expand karunge.
X = Z

Open(E)
f(S-A-Z-E) = (1+4+2) + 0
= 7

ab hamne answer yehi mil gaya par hum sari possibilities check karunge ki best answer yehi hai ki nahi.

Toh ab baki function bhi compare kara
min (f(S-B-C), f(S-B-D), f(S-A-Y))

= min (13, 18, 16)

13 C ka hai toh hum C ko Expand karunge.

X = C
Open (E)

f(S-B-C-E) = 2+7+5
= 14

min (f(S-B-D), f(S-B-Y))
min (18, 16)

Y ko expand karunge

X = Y
Open (Y)
f(S-A-Y-E) = 1+7+3
= 11

X = D

Open(E)

f(S-B-D-E) = 2+1+12
= 15,

Thus sabse best solution hai

f(S-A-Z-E) = 7,

EXPERIMENT : No.

PAGE No.
DATE

Genetic Algorithm

function : GENETIC ALGORITM (population, FITNESS_FN)
return an individual

input : population, a set of individuals
FITNESS_FN, a function that represent
the quality of individual

repeat

new population ← empty

loop for i from 1 to size(population) do

x ← Random_Selection (population, FITNESSFN)

y ← Random Selection (population, FITNESS_FN)

child ← REPRODUCE (x, y)

if (small Random probability) then child ← mutate (child)

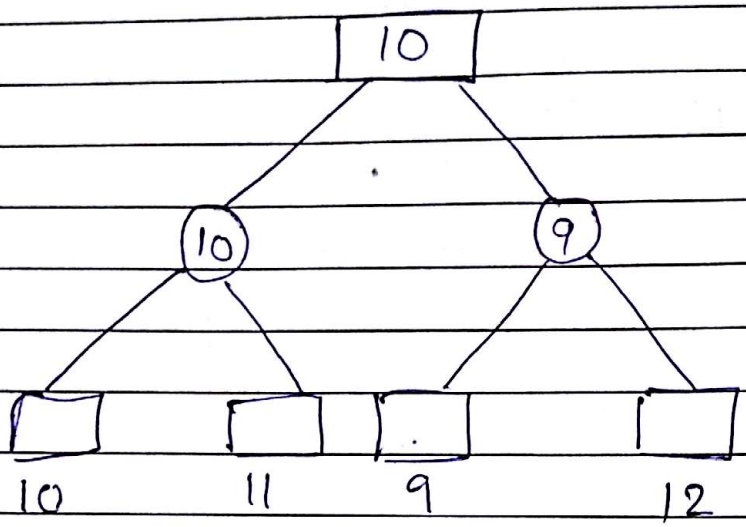
add child to new population

population ← new population

until a fit enough individual is found or enough time has elapsed

return best individual

min-max Algorithm



Initially we will take the worst condition for max and min

$max = -\infty$
$min = \infty$

Now we will start from left node

Now minimum is playing so minimum will be updated.

minimum

1] min(∞, 10) = 10

min(10, 11) = 10

Thus we got 10 for first node of minimum.

2) Now second minimum is playing to the right of it.

min(∞, 9) = 9
min(9, 12) = 9

Thus we got 9 as answer for second minimum's node

MAXIMUM

now maximum is playing thus its value will be updated

max(-∞, 10) = 10 (first node is Reached)

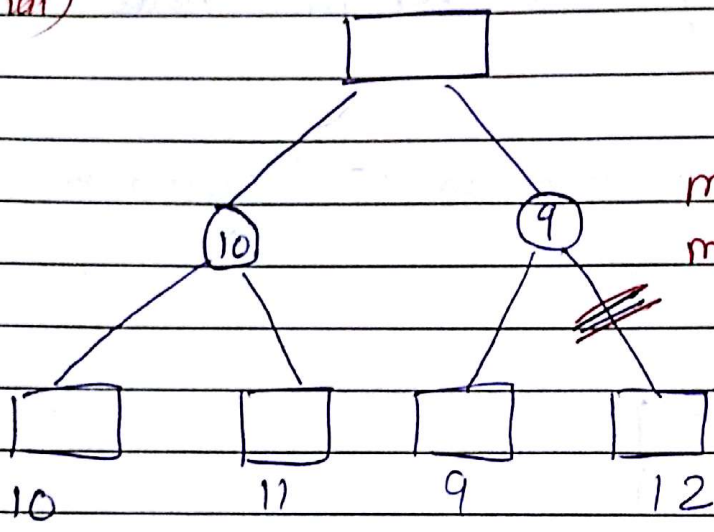
max(10, 9) = 10 (second node is reached)

max = 10

Answers

Alpha beta prunning

(yeah thora bahut alpha beta min max jaisa hi hai bus yeah unwanted nodes ko kaat deta hai)



initial

max (yani) = α
min (yani) = β

$\alpha = -\infty$
$\beta = \infty$

We will start from left node now β is playing so α value i.e ∞ will be constant and not changed only β value will be updated

$\alpha \geq \beta$

prunning

β

$\alpha = -\infty$
$\beta = (\infty, 10)$ ← Read first
$= 10$ node

	$\alpha \geq \beta$	Running
$\beta = (10, 11) = 10$		
ab check karne ka $\alpha \geq \beta (-\infty \geq 10)$ no toh cancel karo uss columns me	X	X
α Now alpha is Playing		
$\alpha(-\infty, 10) = 10$ $\beta(\infty)$	X	X
ab jo yeah alpha ki value hai voh niche ke nodes me update hongi,		
β now β is Playing $\alpha = 10$ $\beta(\infty, 9) = 9$ yeha pe $\alpha \geq \beta$ condition satisfies toh Running hongi	✓	✓

Sundaram

Teacher's Sign.: _____

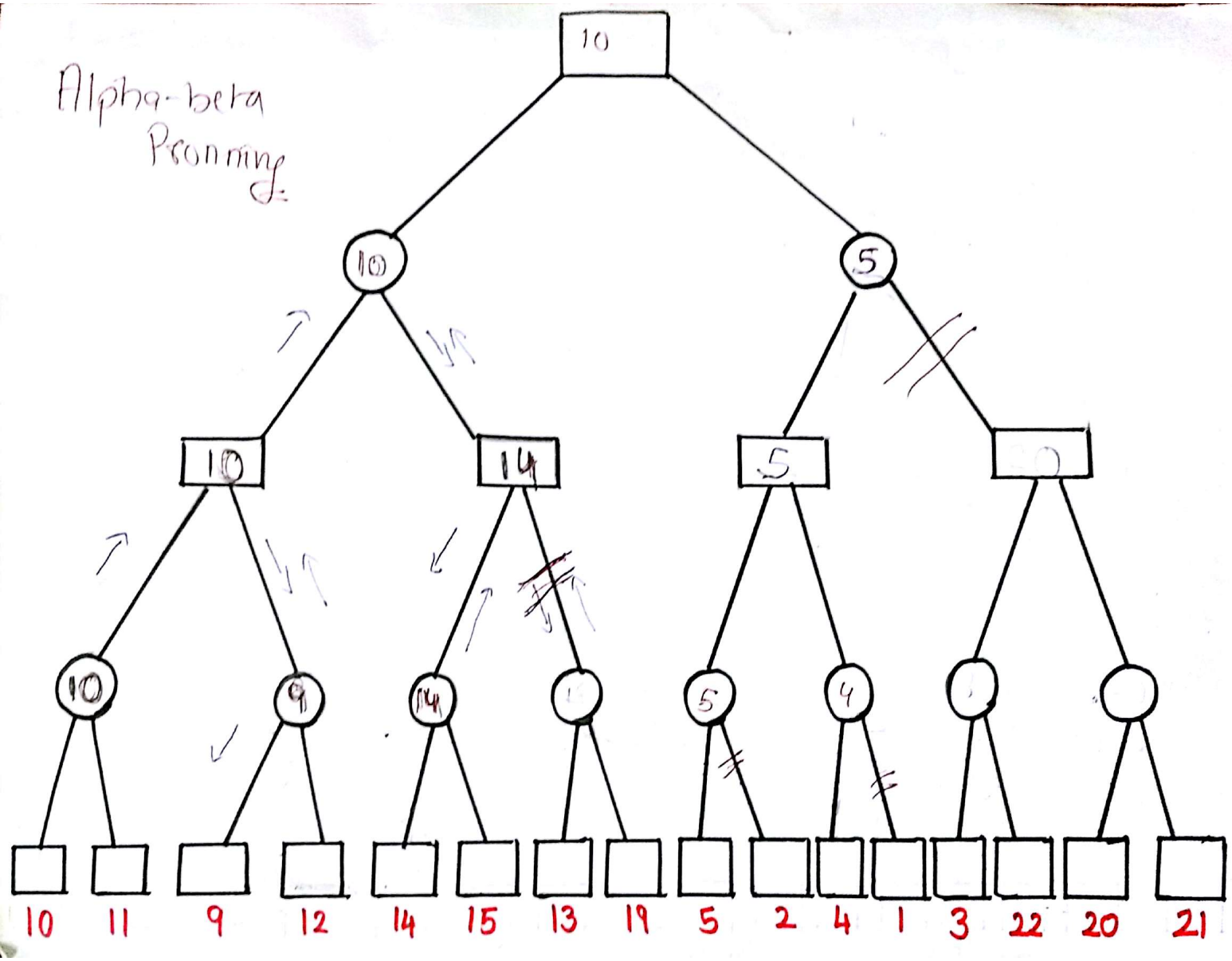
EXPERIMENT : No.

PAGE No.
DATE

α now α is Playing
β value will be fixed
$\beta = \infty$

$\alpha(10, 9) = 10$

Bus answer is
10 only same like
min max but we
have eliminated some nodes.



we will start from left most node		$\alpha \geq \beta$	Pruning
β	is Playing α will be constant or fixed $\alpha = -\infty$ $\beta(\infty, 10) = 10$ $\beta(10, 11) = 10$	X	X
α	is Playing β will be constant or fixed $\beta = \infty$ $\alpha = (-\infty, 10) = 10$ ~~$\alpha = \alpha$~~		
β	is Playing α will be constant but now alpha value will be 10 because upar se alpha ki value niche aai hai (if not understood see the video		

$\alpha \geq \beta$	Pruning
✓	✓
X	X
X	X

$\alpha = 10$

$\beta = (\infty, 9) = 9$

condition sahi hai
12 will be prunned

α is Playing
β will be constant
or fixed

$\alpha(10, 9) = 10$

β is Playing
α will be constant

$\beta(\infty, 10) = 10$
ab yeah niche ke
nodes me jayegi value
$\alpha = -\infty$

β is Playing
α will be constants

$\beta(10, 14) = 10$
$\beta(10, 15) = 10$
$\beta(14, 15) = 14$
β ki value update nahi
hongi.

α ≥ β | Prun

α is playing
β will be constants
i.e 10

α (-∞, 14)
= 14
α ≥ β condition
satisfied

β is Playing
α will be Constant

β (10, 14) = 10

α is Playing
β will be constant

α (-∞, 10) = 10
This α value Baki
sare node me jayengi
upar se niche

β is Playing
α constants ie 10
β (∞, 5) = 5
condition satisfied
α ≥ β

	$\alpha \geq \beta$	Pruning
• α is Playing β constant $\alpha(10,5) = 10$ $\beta = \infty$	X	X
• β is Playing α constant i.e 10 $\beta(\infty, 4) = 4$ $\alpha = 10$ $\beta = 4$ Condition satisfies	✓	✓
• α is Playing β is constant $\alpha(10,5) = 10$ $\alpha(10,4) = 10$ $\alpha(5,4) = 5$ $\beta = \infty$	X	X

EXPERIMENT : Page No: 35

PAGE No.
DATE

- β is Playing α will be constant = 10

  β(∞, 5) = 5

  α = 10, β = 5

  condition satisfies

- α is Playing β will be constants

  α(10, 5) = 10

  α = 10

  Answer

α ≥ β	Running
✓	✓

IF YOU HAVE
ANY DOUBT
YOU CAN EMAIL US

Email: Sumerrajpurohit007@gmail.com

OR

CALL AT: 9762903078