Game search:

- Games are a form of multi-agent environment
    - What do other agent do?
    - How do other agents affect our success?
    - Cooperative vs competitive multi-agent environment
    - Multi-agent competitive environment gives rise to adversarial search often known as game search

Example: Tic-tac-toe

- Two player denoted by:

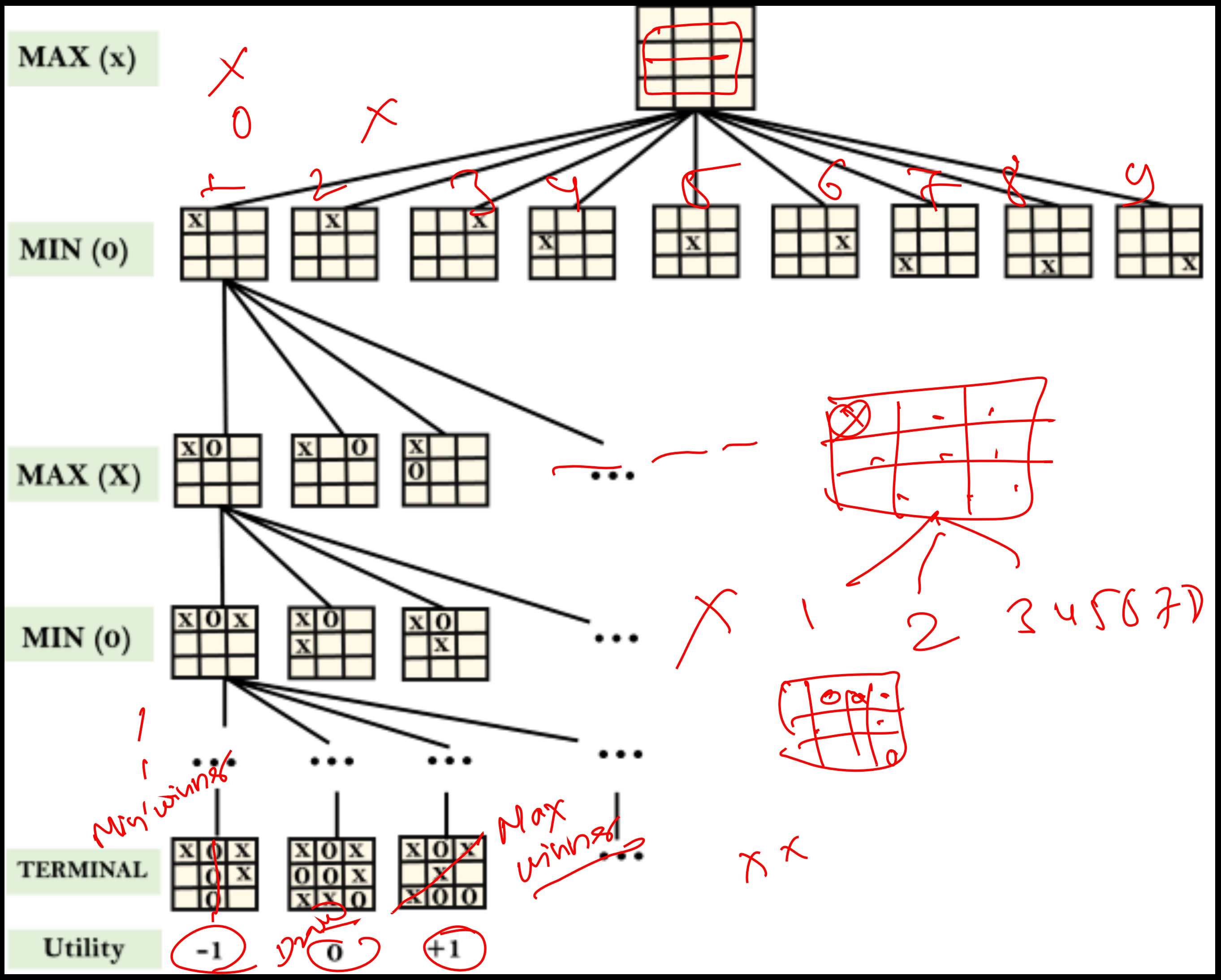
O & X   O X

- 9 cell
- 3x3 board

	O	X
O	X	O
X		

- Winner:

who succeeds in writing 3 letters in a line



partial game tree of Tic-tac-toe

- formal definition of game:

- Initial state

- Successor function: gives a list of (move, state) pair

- Terminal test: determine when the game is over

- Utility functions: gives numerical value of terminal state.

i.e. +1 → win

-1 → loose

0 → draw

Adversarial searching techniques:

- Mostly used in game or decision making situation.
- Outcomes are influenced by the actions of opponent

Types of adversarial search:

- Mini-Max search
- Alpha-Beta pruning

Mini_Max search:

- A kind of backtracking algorithm, used in decision making & game theory.
- used to find optimal move for a player
- Widely used in 2 player based games like tic-tac-toe, chess etc.

- Two players are involved,
  - Maximizer (Max)
  - Minimizer (Min)
- Max tries to get highest possible score where as Min tries to get lowest possible score.

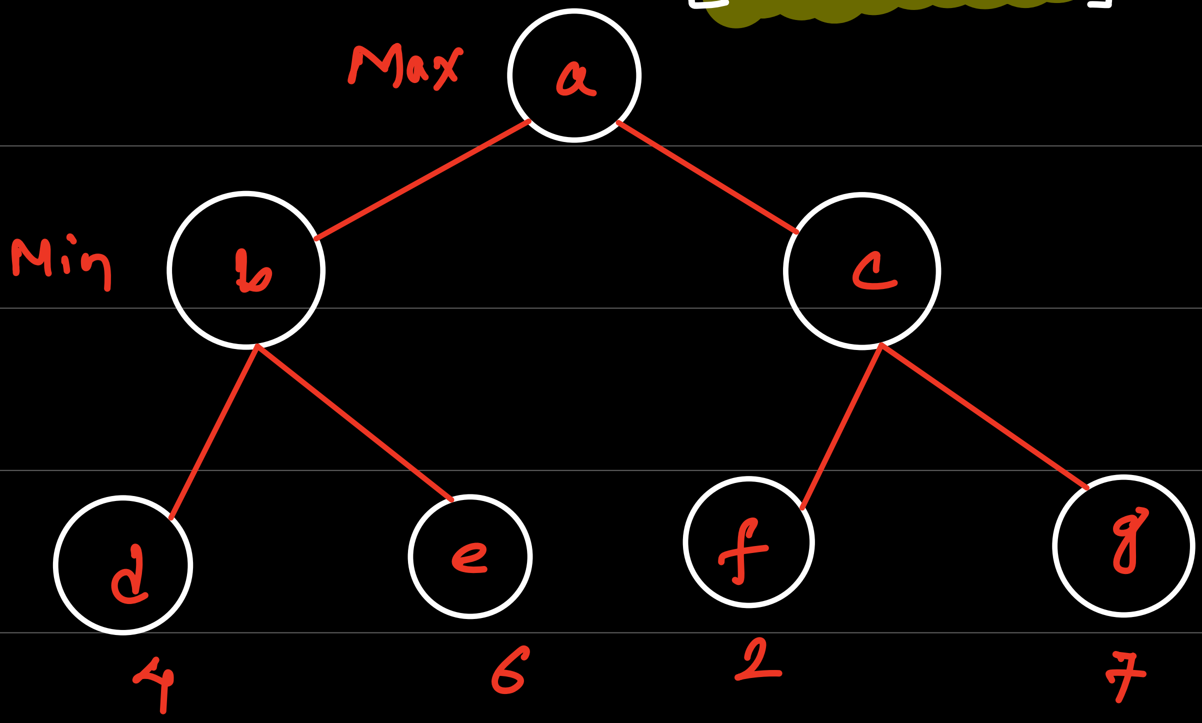
Example:

Consider a game which has 4 final states and path to reach final states are from root to 4 leaves as shown below as a strict binary tree.

Assume you are Max player & you get the first chance to move i.e. you are the root & opponent is at next level.

Which move you would make as a maximizing player considering that your opponent also plays optimally.

[we will start with DFS]

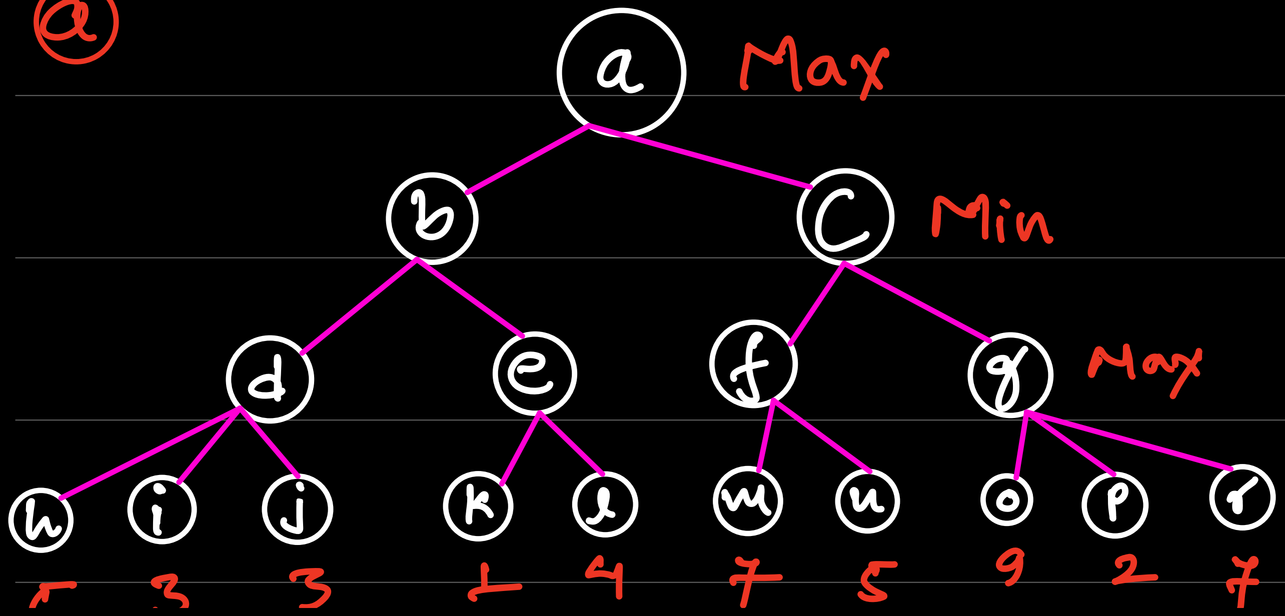


This is a backtracking algorithm, it tries all possible moves, then backtracks & makes a decision.

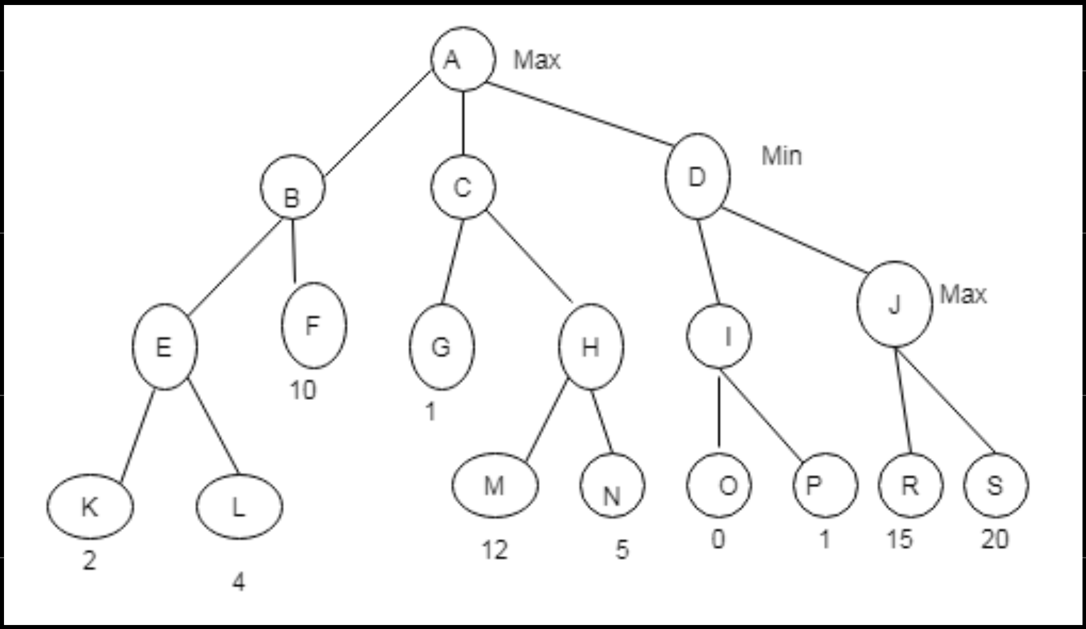
Exercise:

Consider the following game tree. What move should be choosen by the two players, assuming that both are using mini-max procedure.

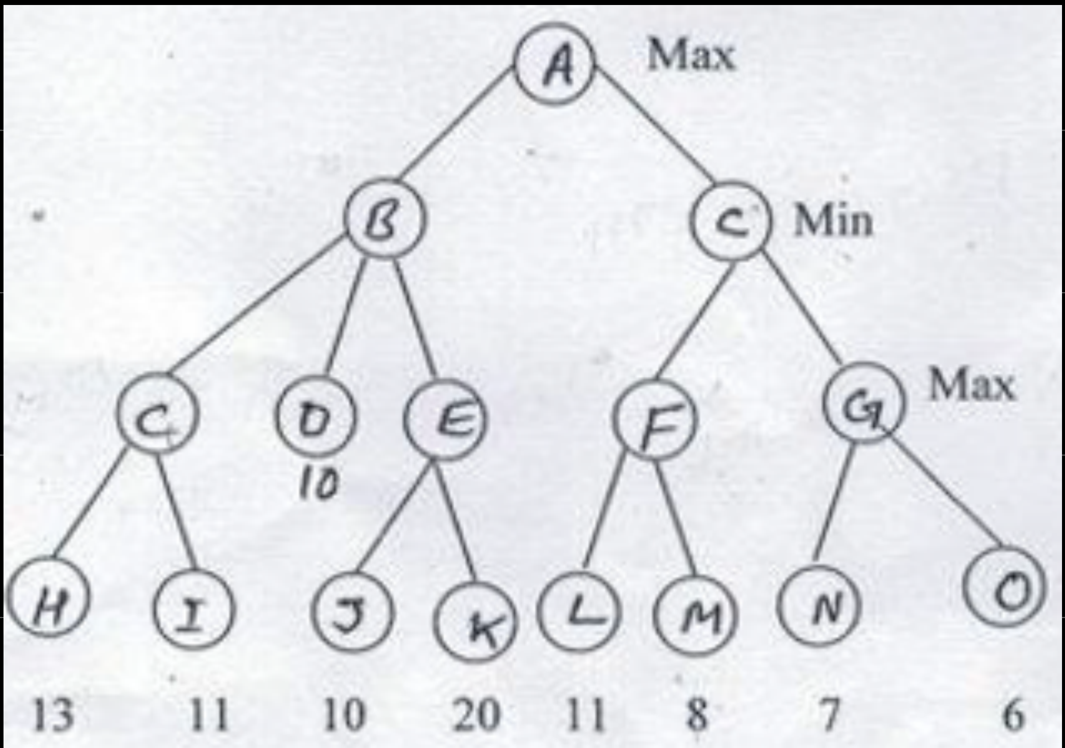
ⓐ



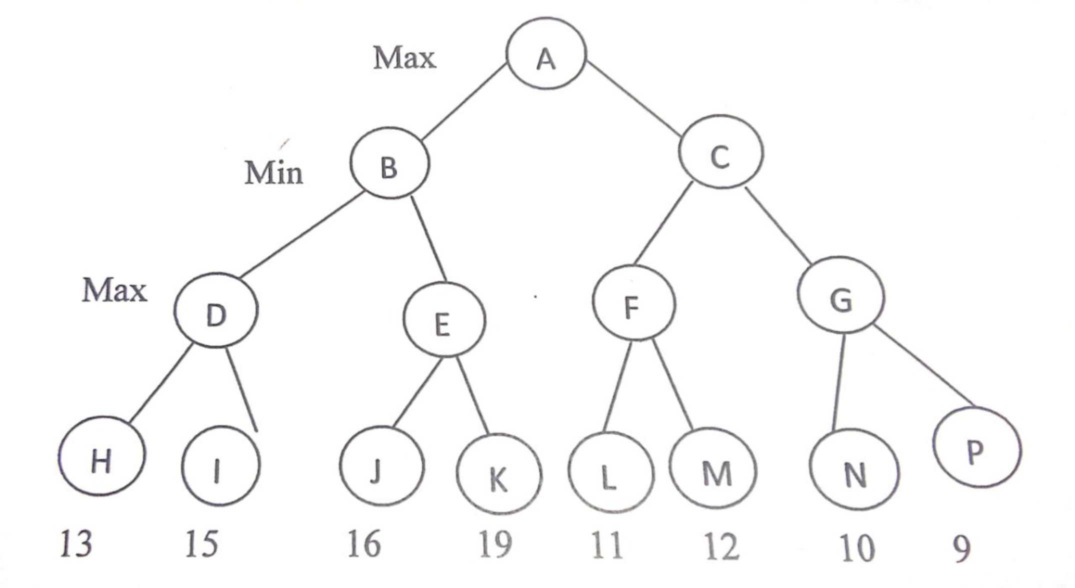
b


A
Max
B
C
D
Min
E
F
10
G
1
H
I
J
Max
K
2
L
4
M
12
N
5
O
0
P
1
R
15
S
20

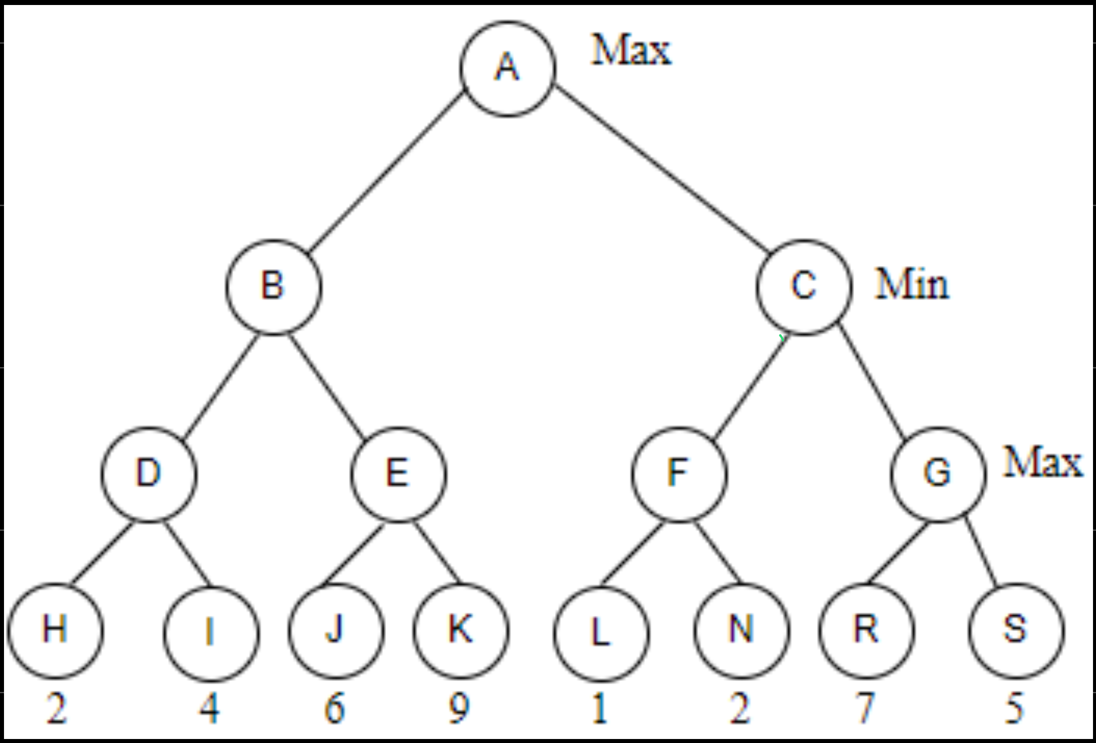
c


A
Max
B
C
Min
C
D
10
E
F
G
Max
H
I
J
K
L
M
N
O
13
11
10
20
11
8
7
6

d


Max
A
Min
B
C
Max
D
E
F
G
H
I
J
K
L
M
N
P
13
15
16
19
11
12
10
9

e


A
Max
B
C
Min
D
E
F
G
Max
H
I
J
K
L
N
R
S
2
4
6
9
1
2
7
5

Alpha-beta purning

- Time complexity of Minimax search is $O(b^d)$
  - b is branching factor
  - d is maximum depth of game search tree

- Because we have to visit every node of the game tree

- Alpha-beta puring is method that reduces the number of nodes explored in minimax search.

- Reduce the time required for search

- Increase perfarmance

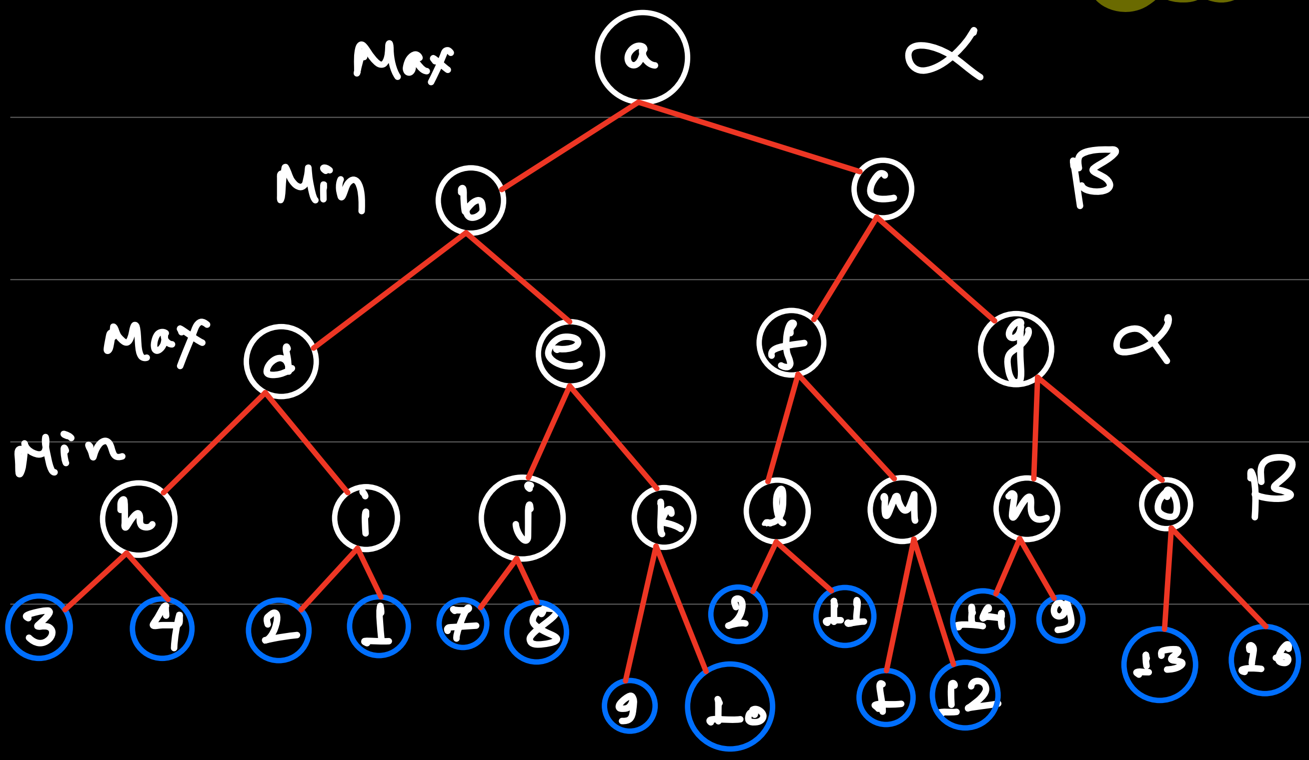
- Here, α & β are two values in game tree where,

  α is consider for Max node

  & β is consider for Min node

Search Strategy
DFS

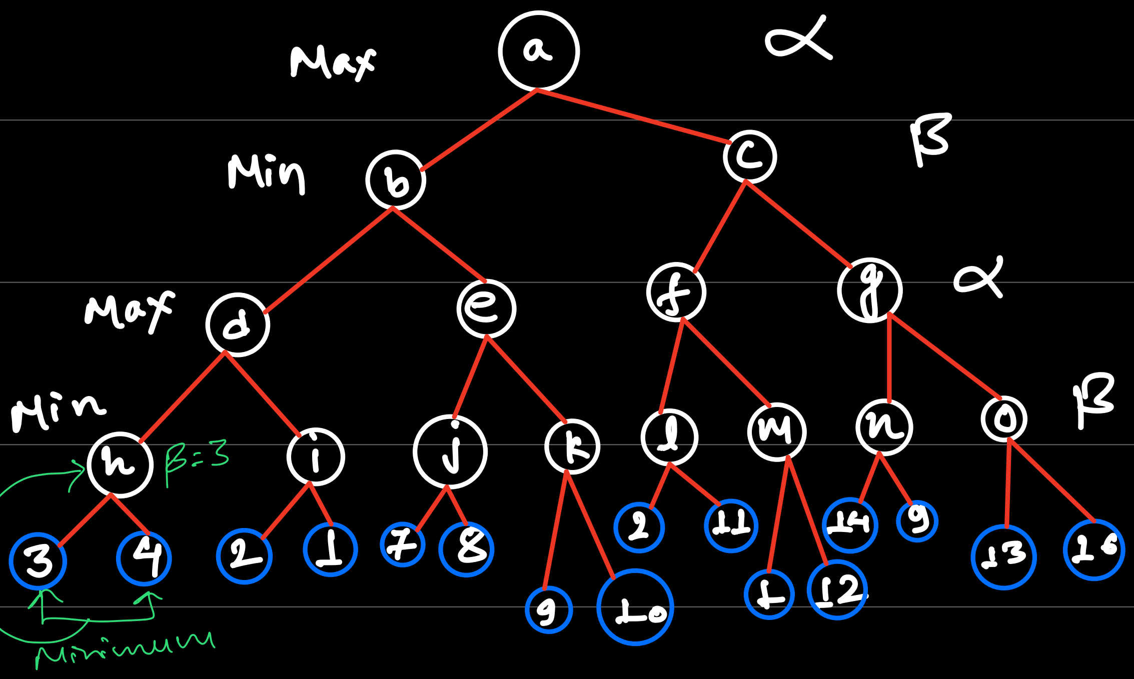
Example



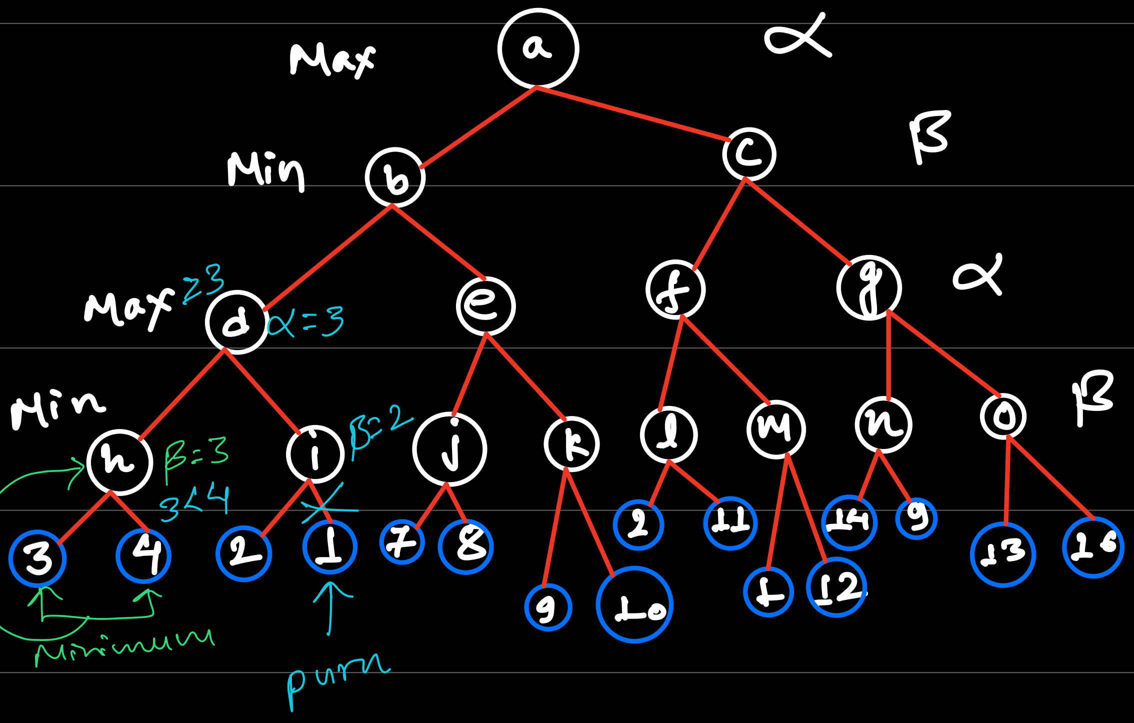
Note: Always value of β is consider in dereasing order from +∞, as upper bound.

Step 1:

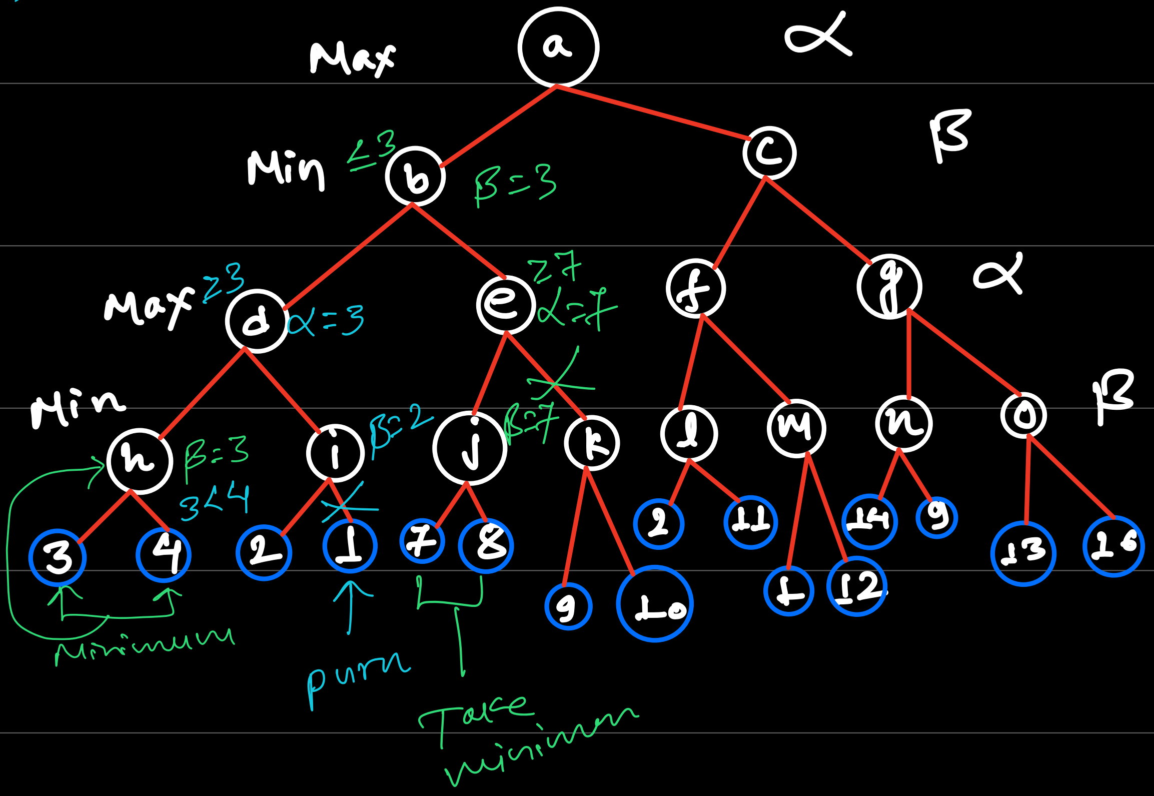
where as, value of α is considre in increasing order form -∞, as lower bound.


Max
a
α
Min
b
c
β
Max
d
e
f
g
α
Min
h
β=3
i
j
k
l
m
n
o
β
3
4
2
1
7
8
9
10
2
11
1
12
14
9
13
16
Minimum

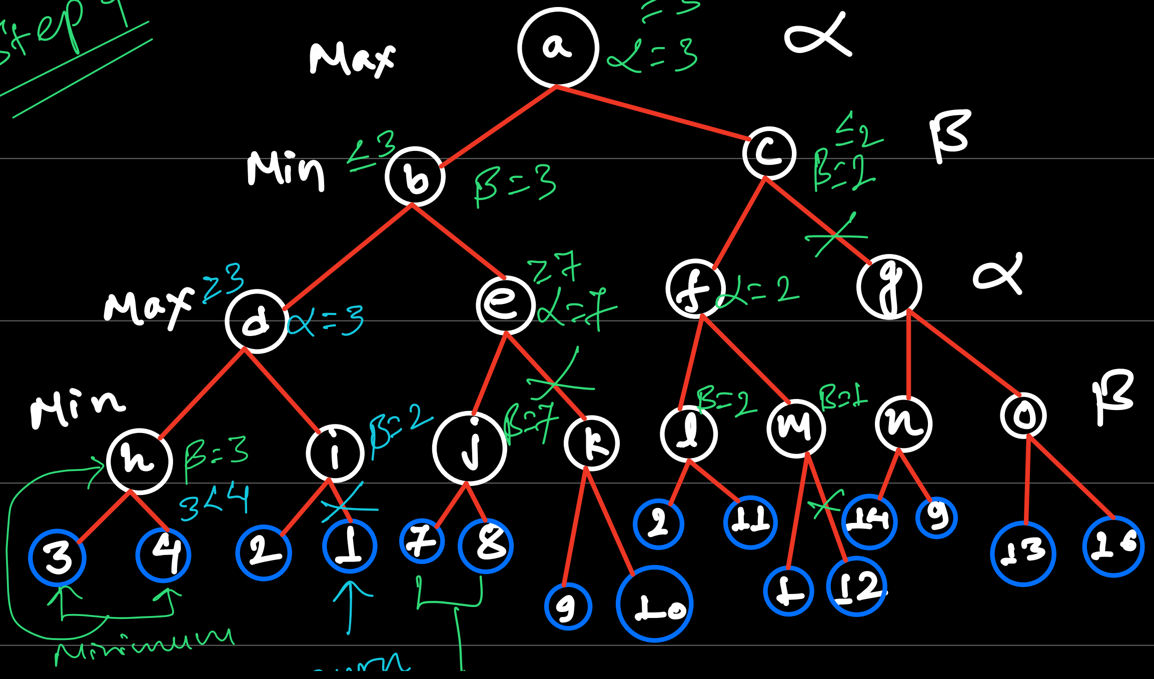
Step 2:


Max
a
α
Min
b
c
β
Max ≥3
d
α=3
e
f
g
α
Min
h
β=3
3<4
i
β=2
j
k
l
m
n
o
β
3
4
2
1
7
8
9
10
2
11
1
12
14
9
13
16
Minimum
prun

Step 3


Max
a
α
Min
≤3
b
β=3
c
β
Max
≥3
d
α=3
e
≥7
α=7
f
g
α
Min
h
β=3
3<4
i
β=2
j
β=7
k
l
m
n
o
β
3
4
2
1
7
8
9
10
2
11
1
12
14
9
13
16
Minimum
purn
Take minimum

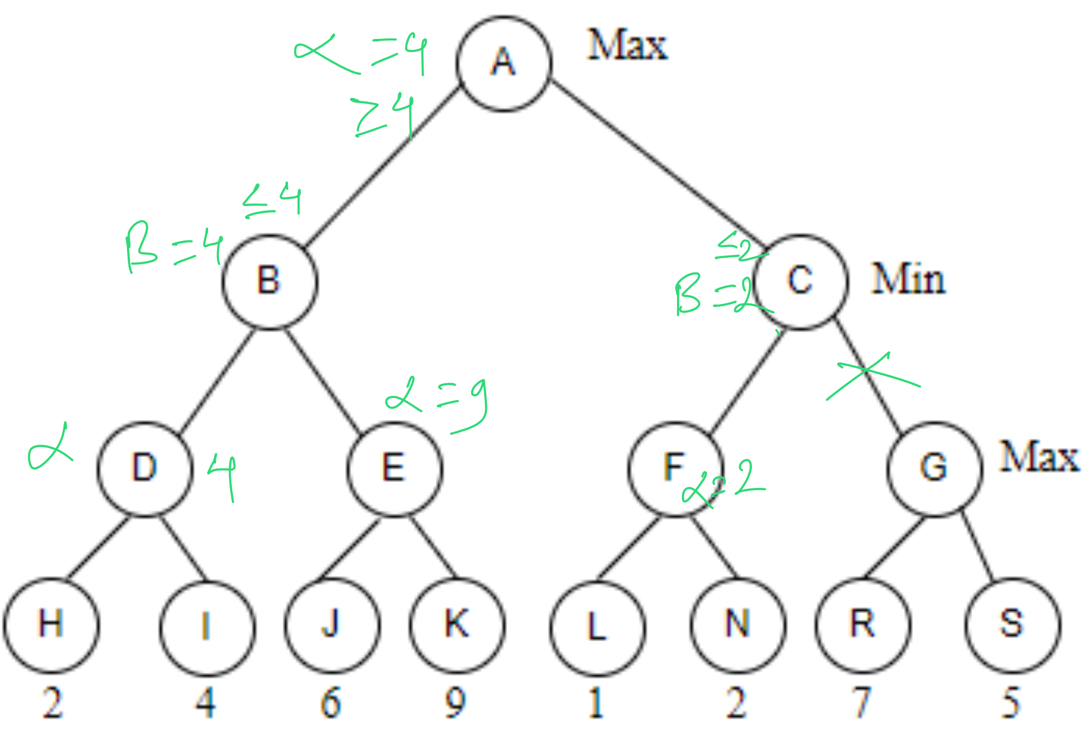
Step 4


Max
a
≥3
α=3
α
Min
≤3
b
β=3
c
≤2
β=2
β
Max
≥3
d
α=3
e
≥7
α=7
f
α=2
g
α
Min
h
β=3
3<4
i
β=2
j
β=7
k
l
β=2
m
β=1
n
o
β
3
4
2
1
7
8
9
10
2
11
1
12
14
9
13
16
Minimum
purn

pu

Take minimum

Example


A
Max
α=4
≥4
B
≤4
B=4
C
Min
≤2
B=2
D
α
4
E
α=9
F
α≥2
G
Max
H
I
J
K
L
N
R
S
2
4
6
9
1
2
7
5