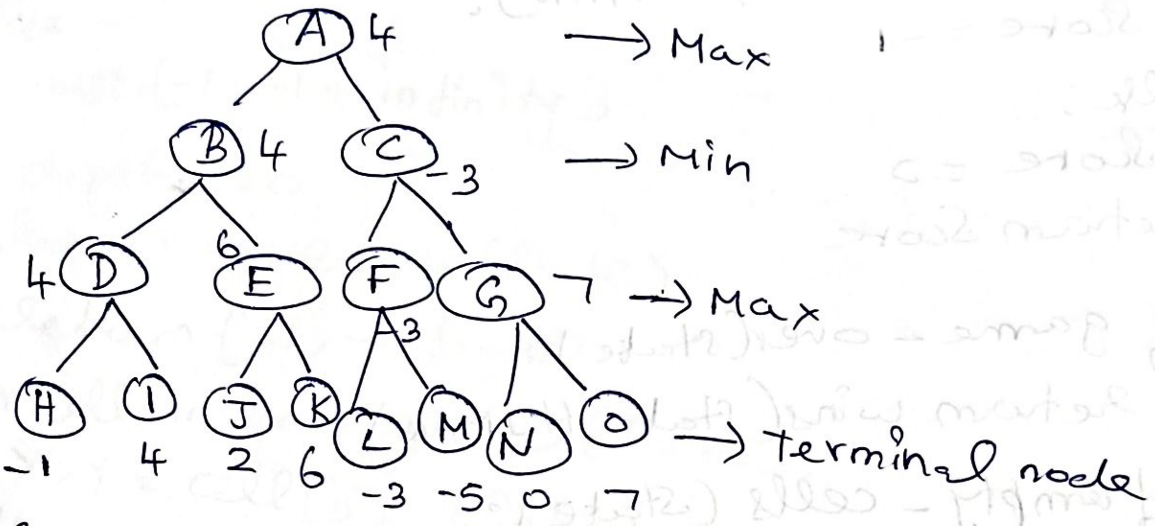
Ex.No:5 Minimax Algorithm

<u>Aim</u>:- Implement Minmax Algorithm in python

<u>Algorithm</u>:-



i) The function recursively evaluate a tree
ii) It takes node depth, depth of tree and a boolean if player is max
iii) If its a terminal node return node value
iv) The function gets childs node asked get child node functions.
v) Compute best score for maximin A.

<u>Soure code</u>:

```
form math import inf as infinity
from random import choice
import platform
import time
form os import system
HUMAN = -1
COMP  = +1
board = +1
 [0, 0,0],
 [0,0,0],
]
def evaluate(state);
 if wins (state , COMP):
   score = +1
 elif wins (state, HUMAN):
   score = -1
 else:
   score = 0
  return score

 def game_over(state):
   return wins(state, HUMAN)
 def empty_cells (state):
   cells = []
   for x, row in enumerate(state);
    for y, cell in enumerate(row);
     if cell = 0.
       cell . append([x, y])
   return cells
  def valid_move (x, y):
     if [x, y] in empty_cells (board]:
       return True
    else :
      return False

  def set_moves(x, y, player):
    if valid_move(x, y):
    board[x][y] = player
    return True
    else:
      return False
```

```
def minimax (State, depth - player).
  if player== = comp :
     best == comP:
      best=[-1 -1, -infinity ]
   else :
      best=[-1, -1 + indinity ]
  if depth = =0
    Score = evaluate (State)
    Return [-1, -1, Score ]
  for cell in empty_cells (State).
    x, y = cell [0], cell (1)
   State [x][y] = player.
   Score = minimax (State, depth-1, Player)
   State [x] [y] = 0
   Score [0], Score [1] = x, y
   if player == COMP :
     if Score [2] > best [2].
        best = Score # max
   else :
        best = Score # min

  Return best

def clear() :
 os - name = platform . system(). lower().
 if windows' in os - name .
    system ('cls')
 else :
    System(' clear')
```

```
def render(state, c_choice, h_choice):
  chars = {
      -1: h_choice,
      +1: c_choice,
      0: ' '
  }
  str_line = '   '
   print('\n' + str_line)
    for row in state:
     for cell in row:
       system = chars [cell]
       print(f'| {system} |' end='')
     print('\n' + str_line)
   }
  ]
```

Output:

```
Chose x or o
chose : x
First to start [y/n] : y
Human turn [x].
 _ _ _ _ _ _ _
 | || _ || |
 | ||   || |
 _ _ _ _ _ _ _
Computer turn [o]
```

Result:-

Thus, The minimax algorithm has been implemented in python.