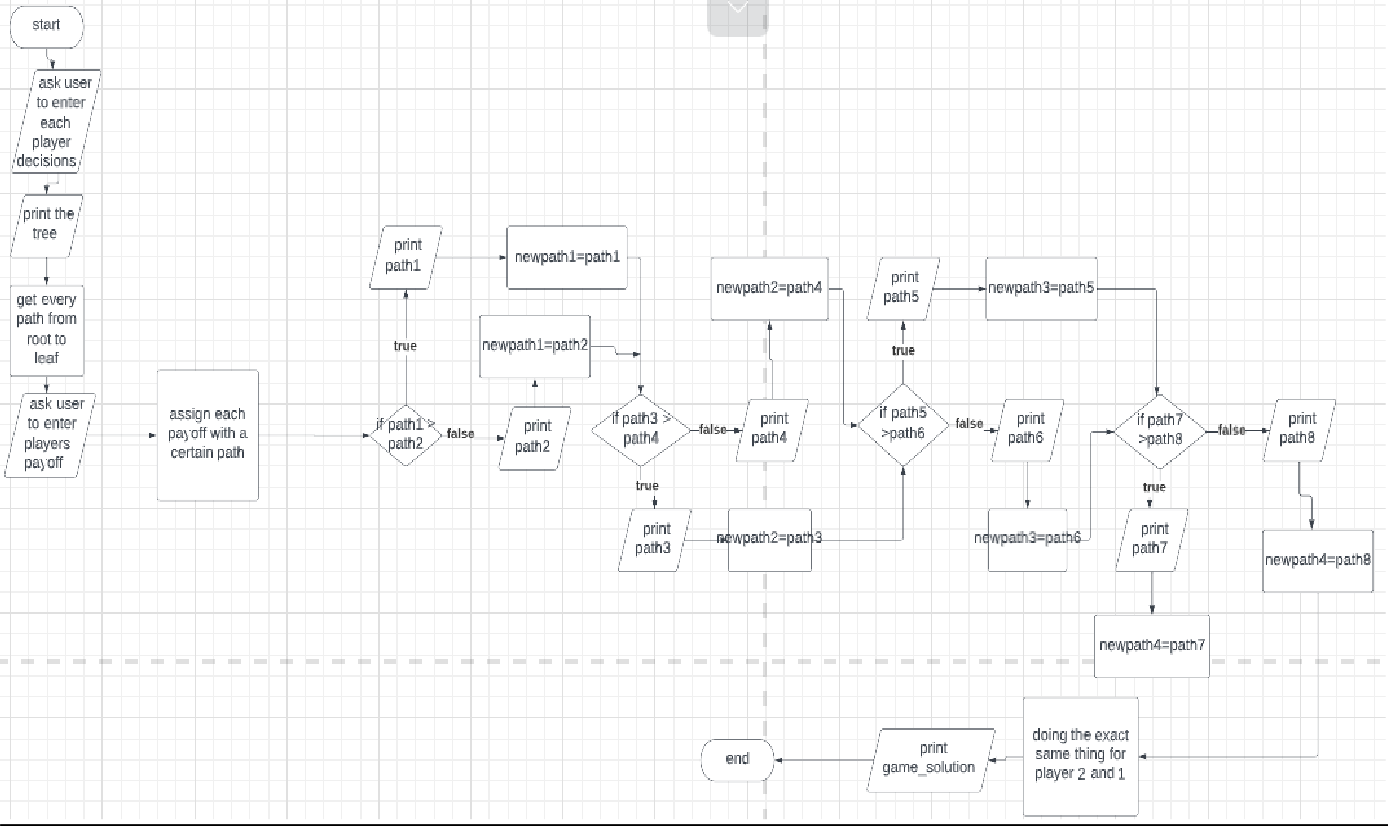
GT-Project Part 3

In this part we will solve a 3-player sequential game where each player has two actions at each decision node.

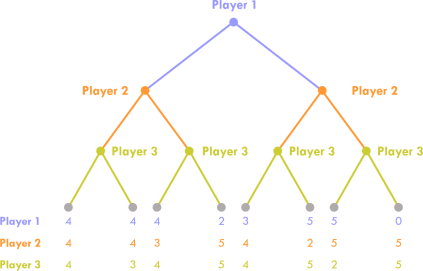
In the following steps our program will be explained:

* First, we will start by implementing a tree using treelib library.
* Ask the user to input the actions for each player.
* In get\_path function we use a dictionary to link each action with its id in the tree then we get all the paths from root to leaf.
* In players\_payoff function we ask the user to input each player payoff for a specific path then we append these payoffs to its path.
* In backward \_induction function we start to compare player 3 payoffs to get only four paths, then we compare player 2 payoffs to get only 2 paths, then we compare player 1 payoffs to get only one path and that is our game solution.

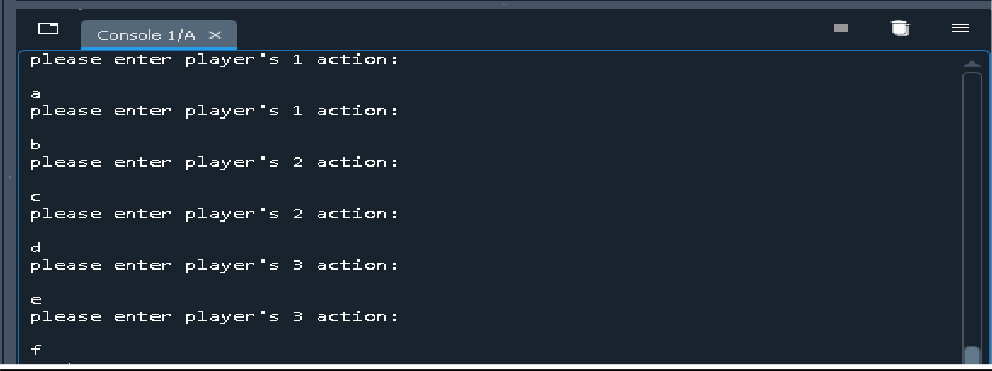
Here’s a flowchart explaining the steps:



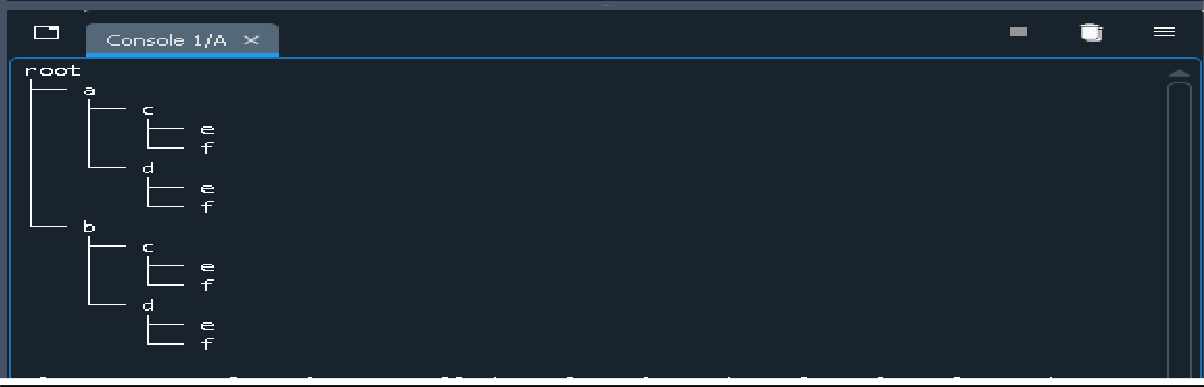
Example1:



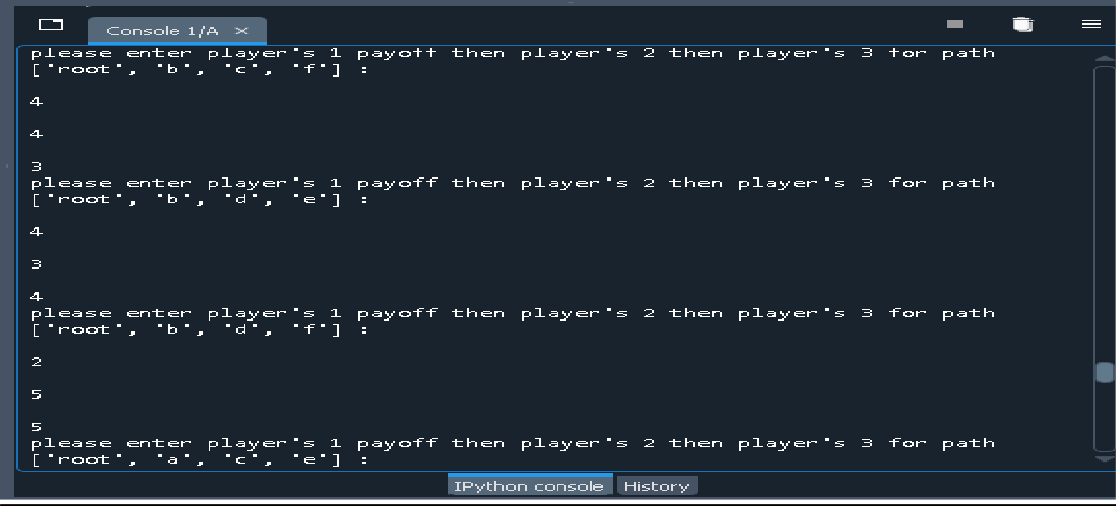
Here’s the input of each player’s decision:

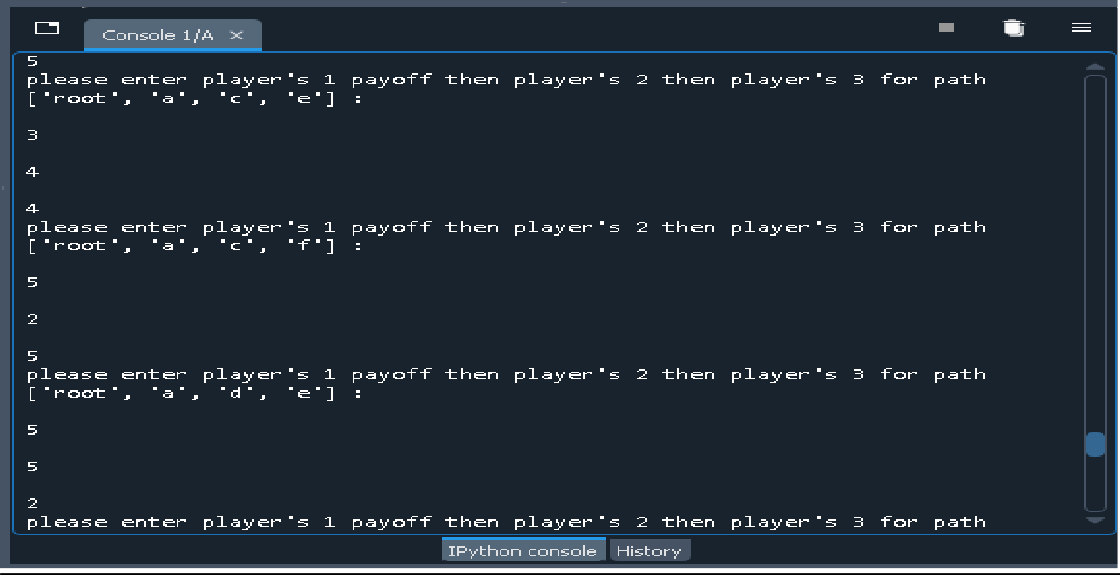


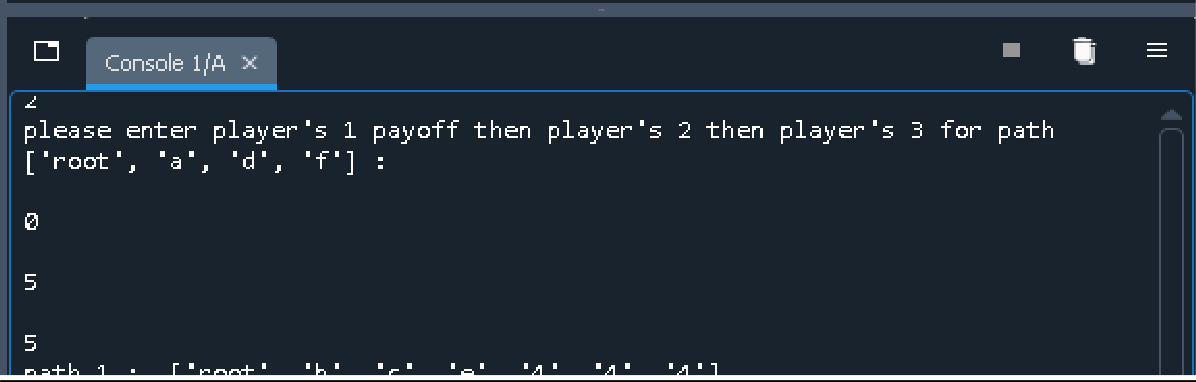
Here’s the tree:



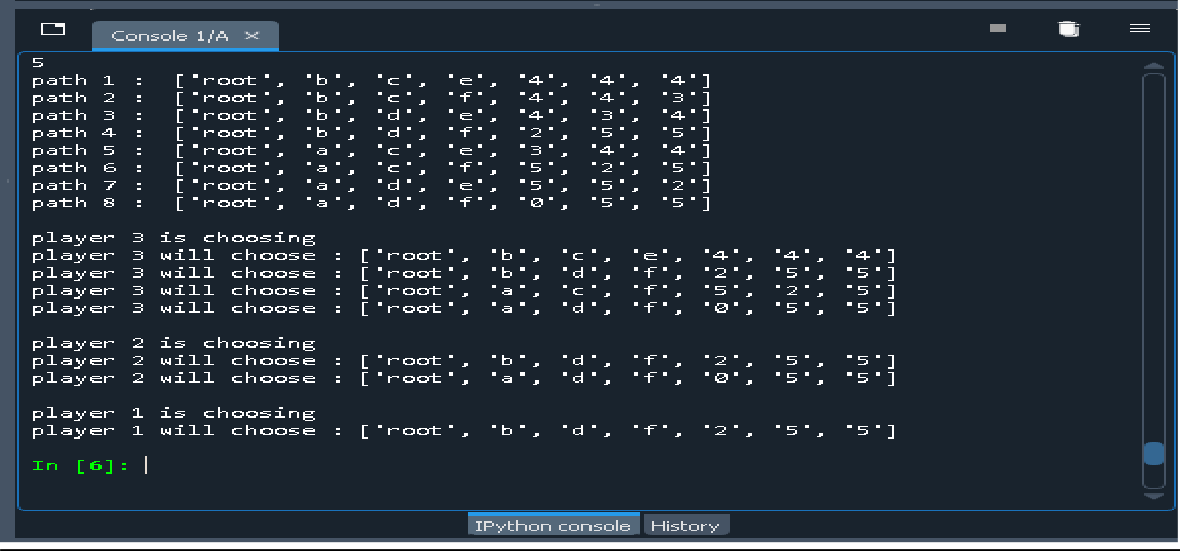
User entering each player payoff:



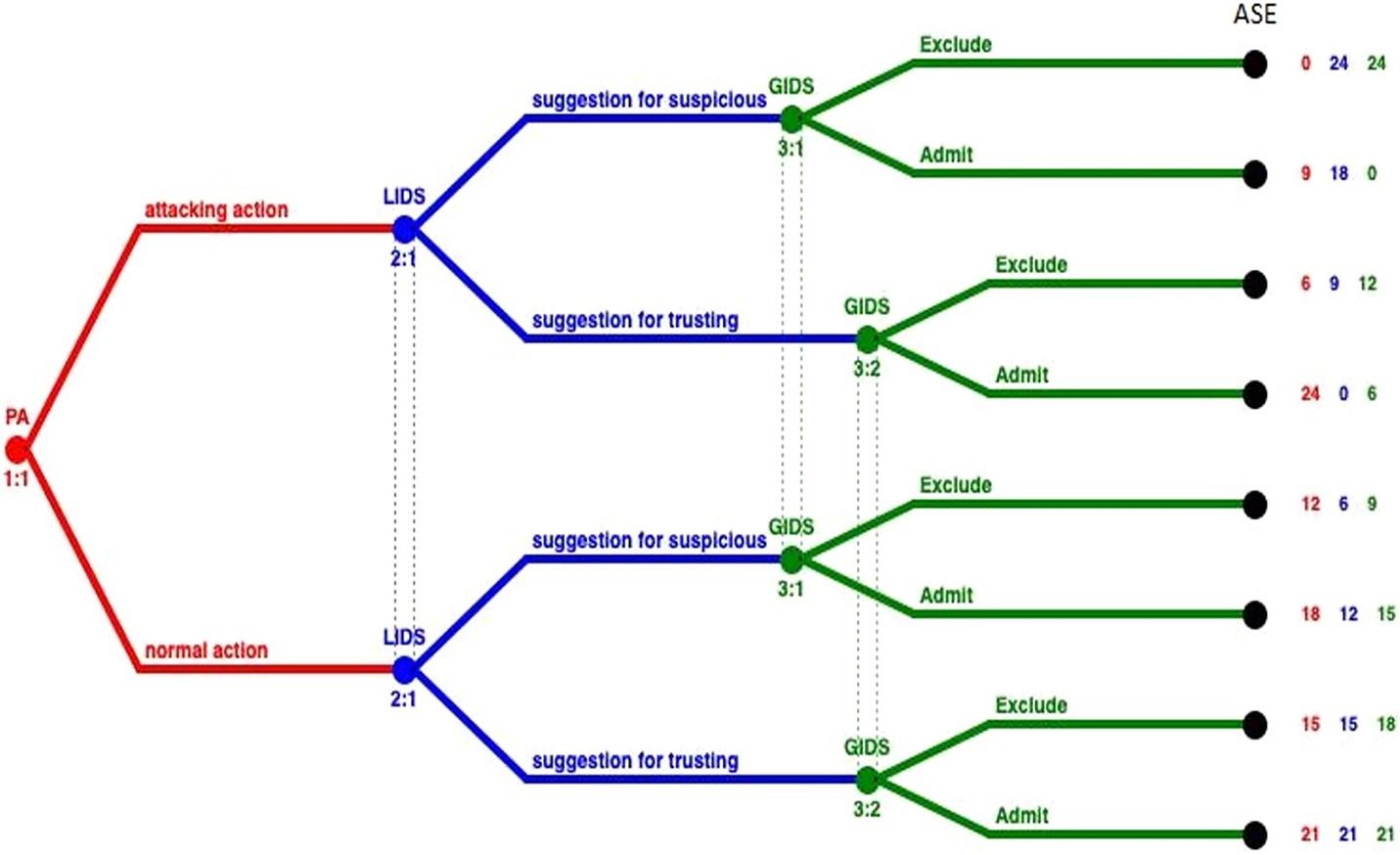




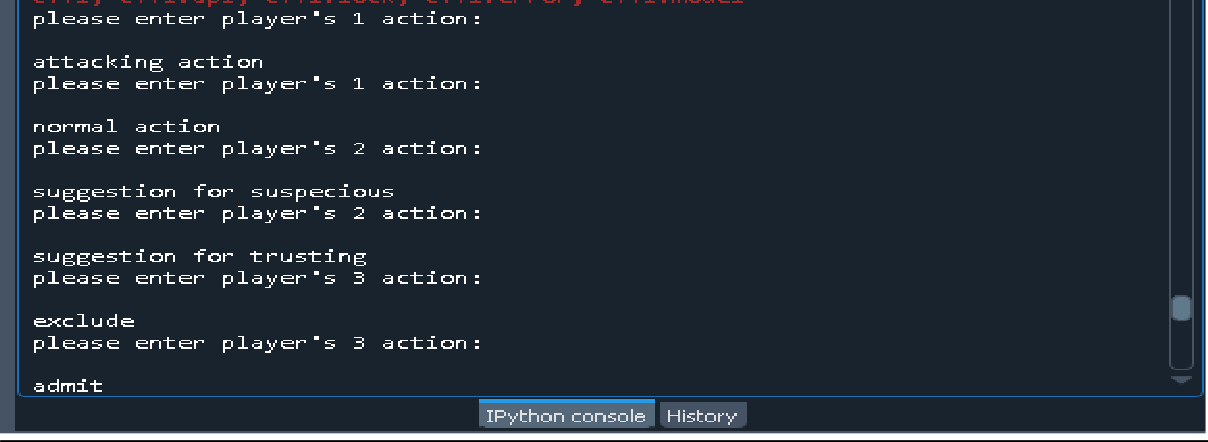
Printing the game solution using backward induction:



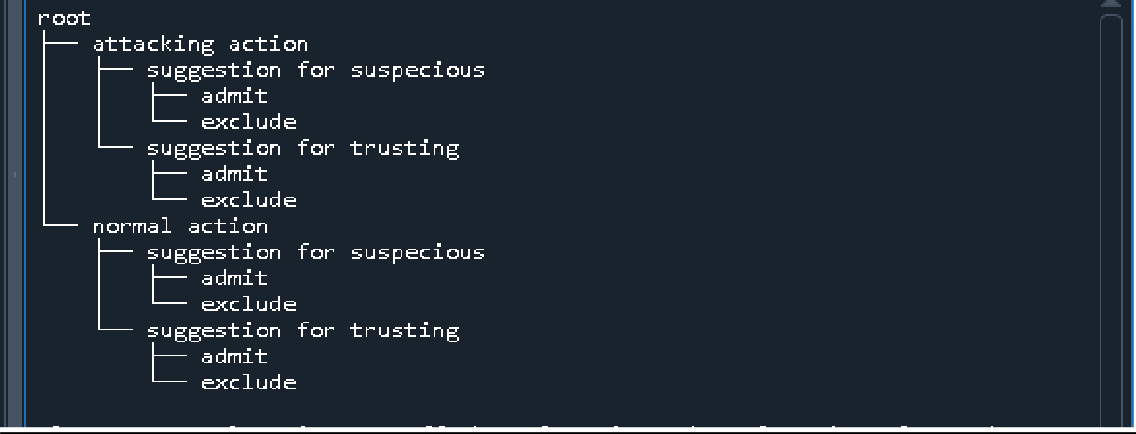
Example 2:



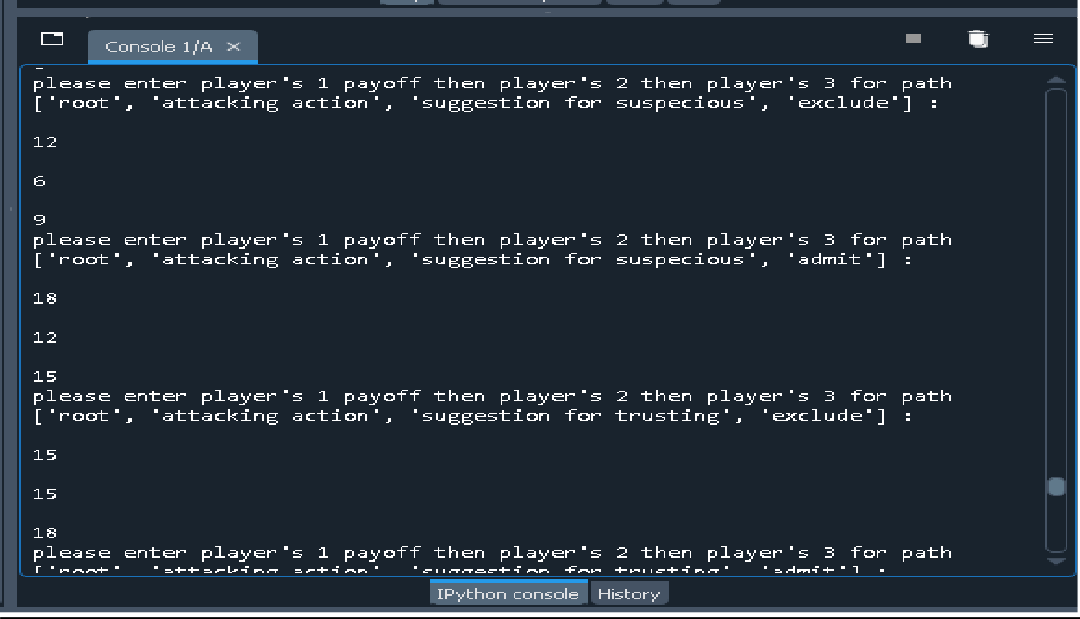
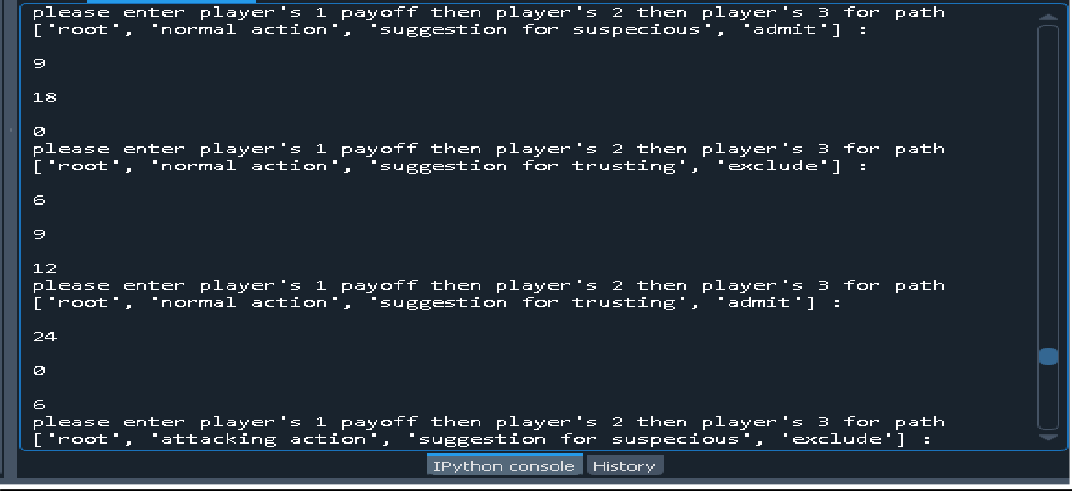
Here’s the input of each player’s decision:

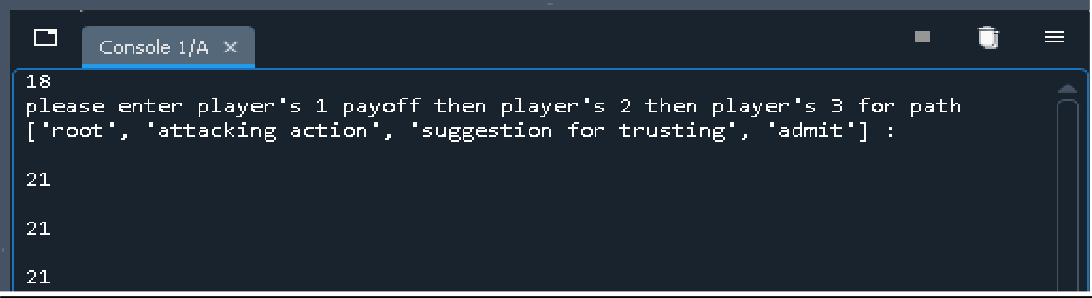


Here’s the tree:

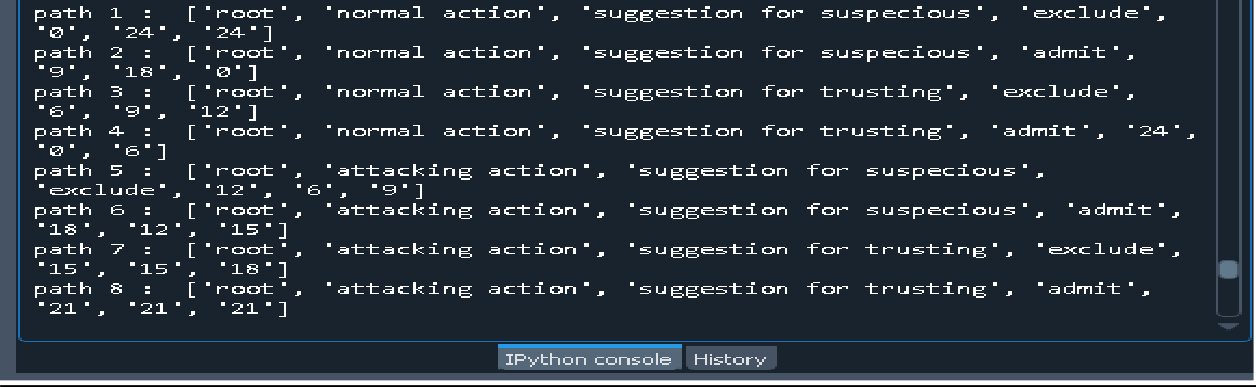


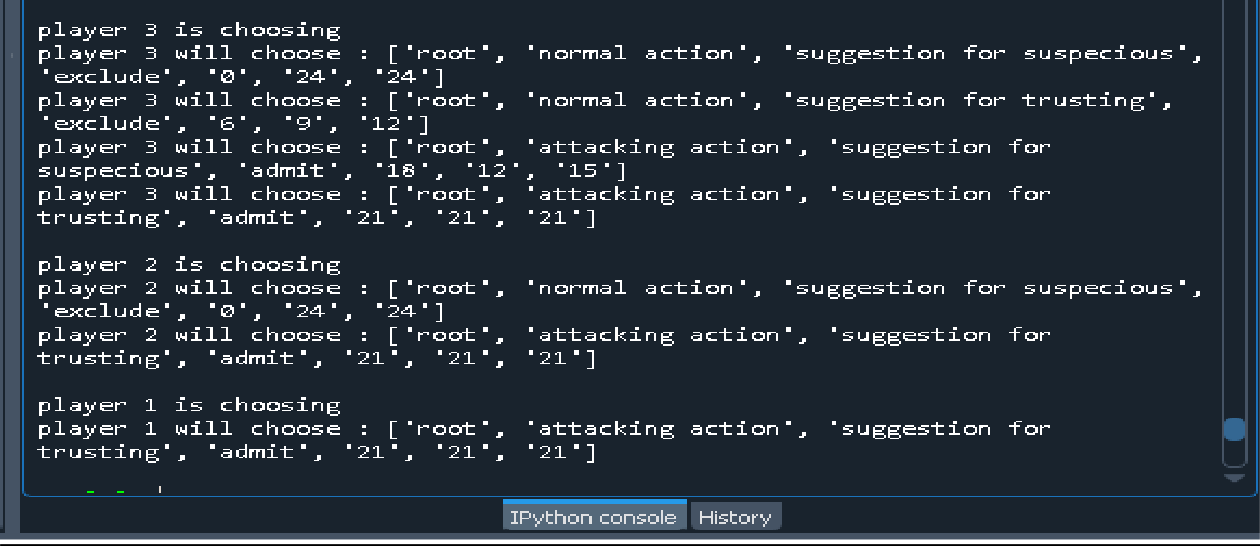
User entering each player payoff:



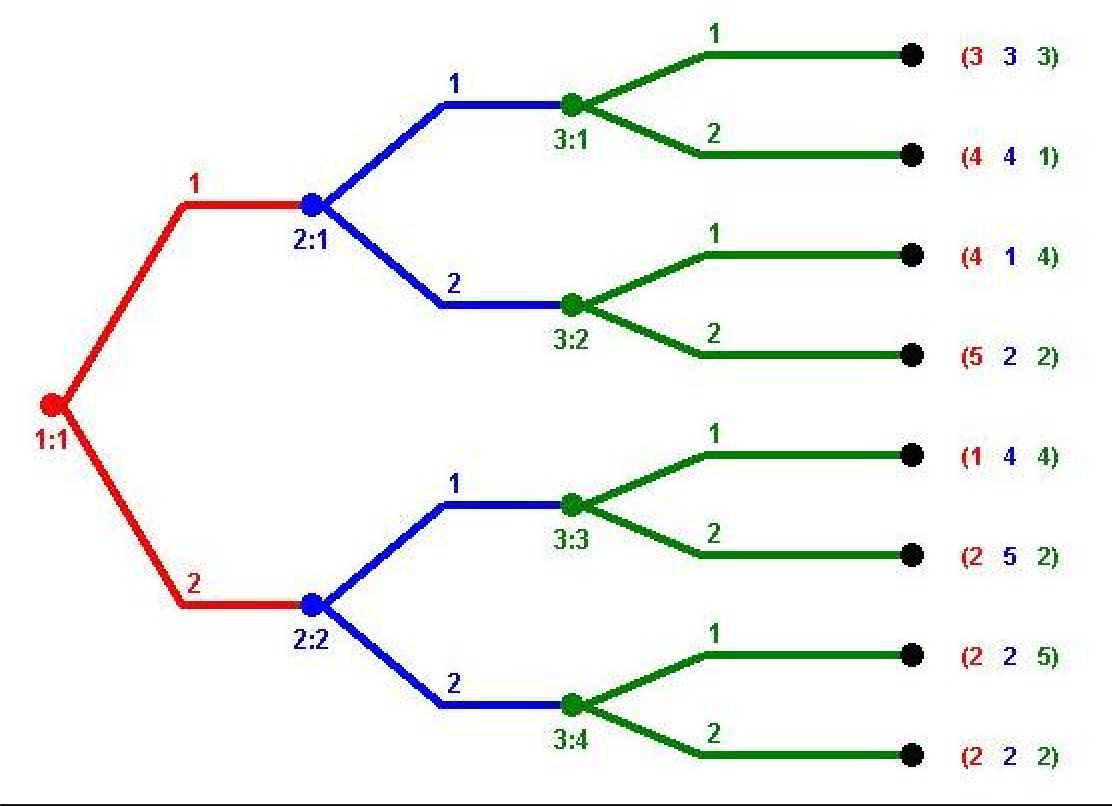


Printing the game solution using backward induction:

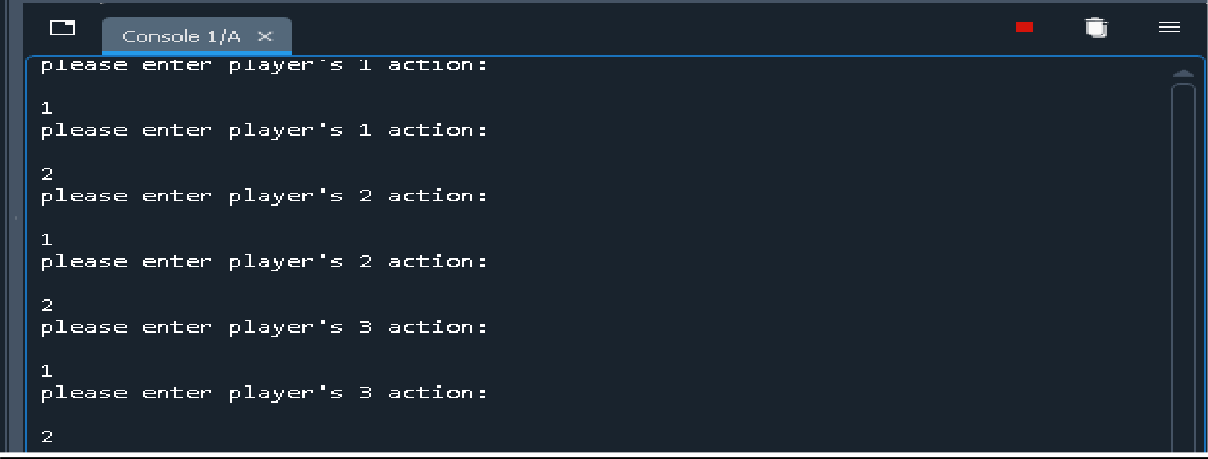




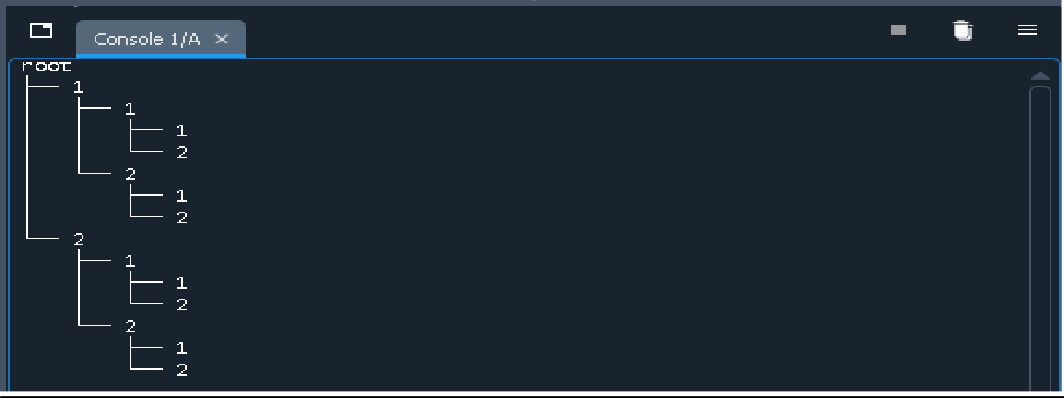
Example 3:



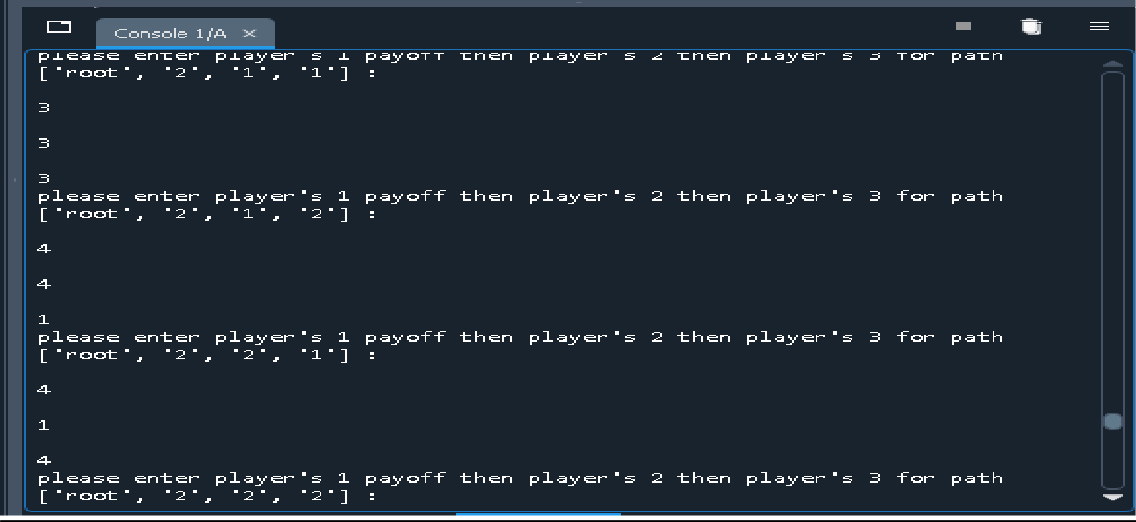
Here’s the input of each player’s decision:

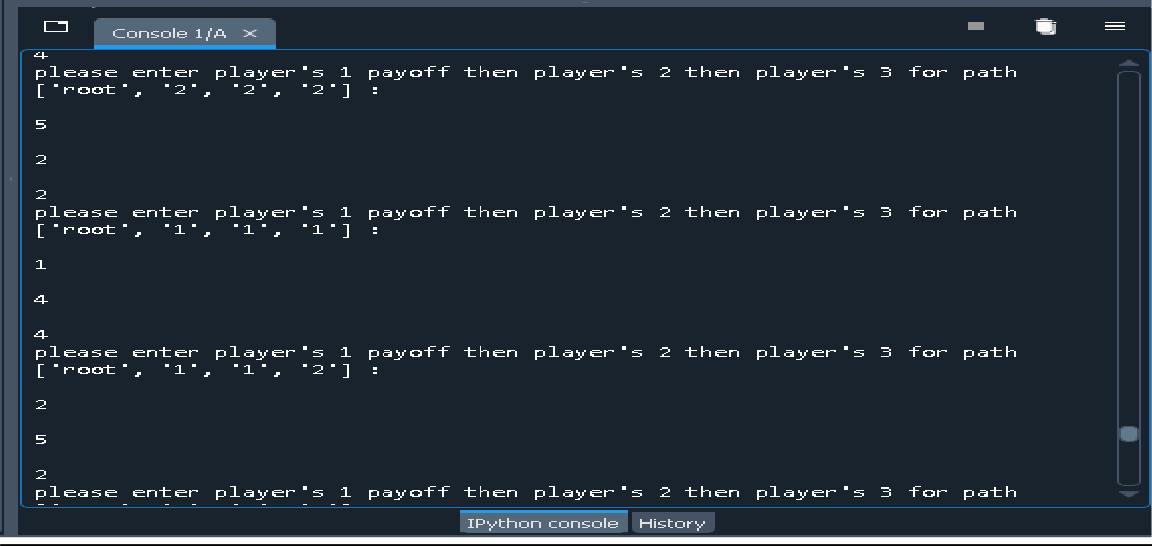


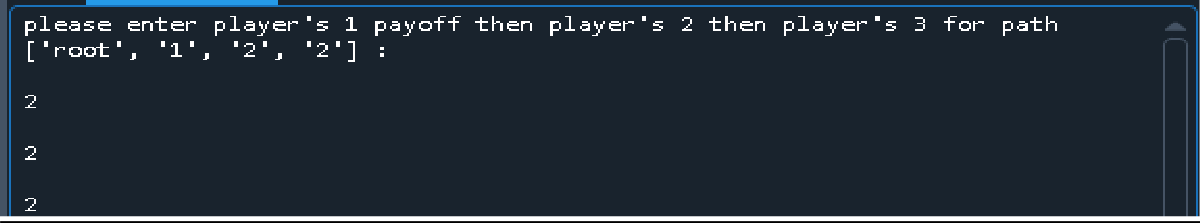
Here’s the tree:



User entering each player payoff:







Printing the game solution using backward induction:

