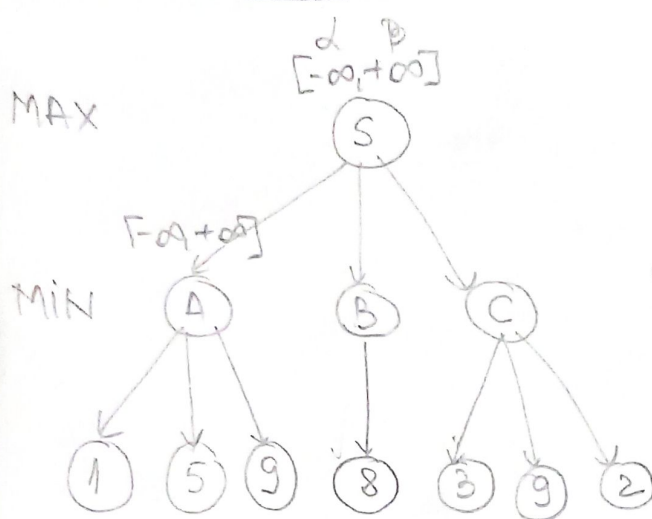


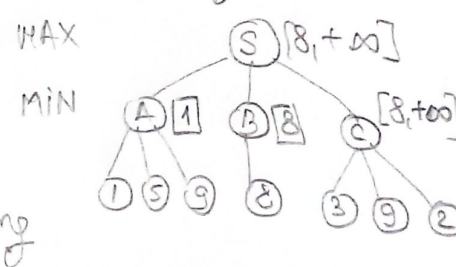
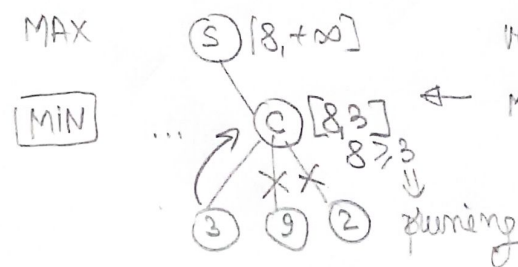
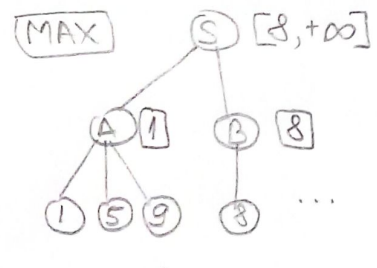
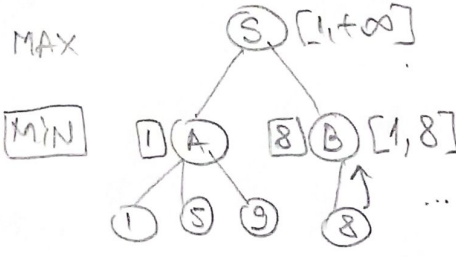
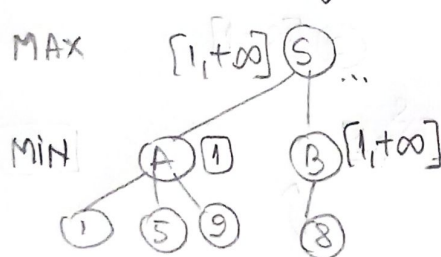
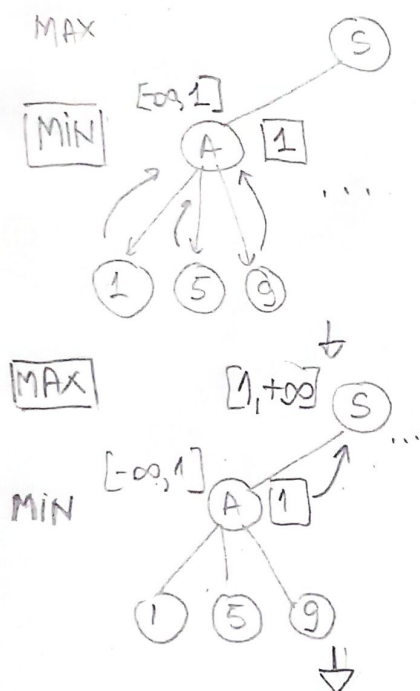


α - β pruning

Initial $\alpha = -\infty, \beta = +\infty$
 MAX maximizes α
 MIN minimizes β

```
def minimax(state, depth, isMaxPlayer,  $\alpha, \beta$ ):
    if (depth == 0 || isLeaf(state)):
        return R(state)

    if (isMaxPlayer):
        value =  $-\infty$ 
        for each valid child of state:
            value = max(value, minimax(child, depth-1, 0,  $\alpha, \beta$ ))
             $\alpha = \max(\alpha, value)$ 
            if  $\alpha \geq \beta$  break # pruning
        return value
    else:
        value =  $+\infty$ 
        for each valid child of state:
            value = min(value, minimax(child, depth-1, 1,  $\alpha, \beta$ ))
             $\beta = \min(\beta, value)$ 
            if  $\alpha \geq \beta$  break # pruning
        return value
```



Explicatie: MAX stie ca poate obtine minim 8 mergand in starea B din S.
 Totodoba, MAX stie ca facand alegerea C va putea obtine cel mult valoarea 3 (deoarece din C MIN o va alege pe cea minima). De aceea, MAX nu mai este interesat sa exploreze si restul de subarbori corespunzatori lui C.