

les algorithmes *MiniMax* et alpha-beta

Un *jeu à deux joueurs* est défini classiquement comme un arbre qui a comme noeuds des *positions* (la racine est appelée la «position initiale du jeu»).

Chaque noeud est un noeud «joueur» (i.e. c'est au joueur de jouer son coup) ou un noeud «opposant», et les joueurs s'alternent, de telle sorte que un noeud joueur a comme fils une liste de noeuds opposants, que l'on obtient à partir des différents coups disponibles (symétriquement pour opposant).

Si un noeud n'a pas de fils, c'est un noeud terminal, et dans ce cas on associe à ce noeud une *valeur* qui indique ce que le joueur a gagné ou perdu.

Typiquement, si on s'intéresse seulement à des jeux comme les échecs, on aura +infini pour gain, -infini pour perte et 0 pour les autres positions.

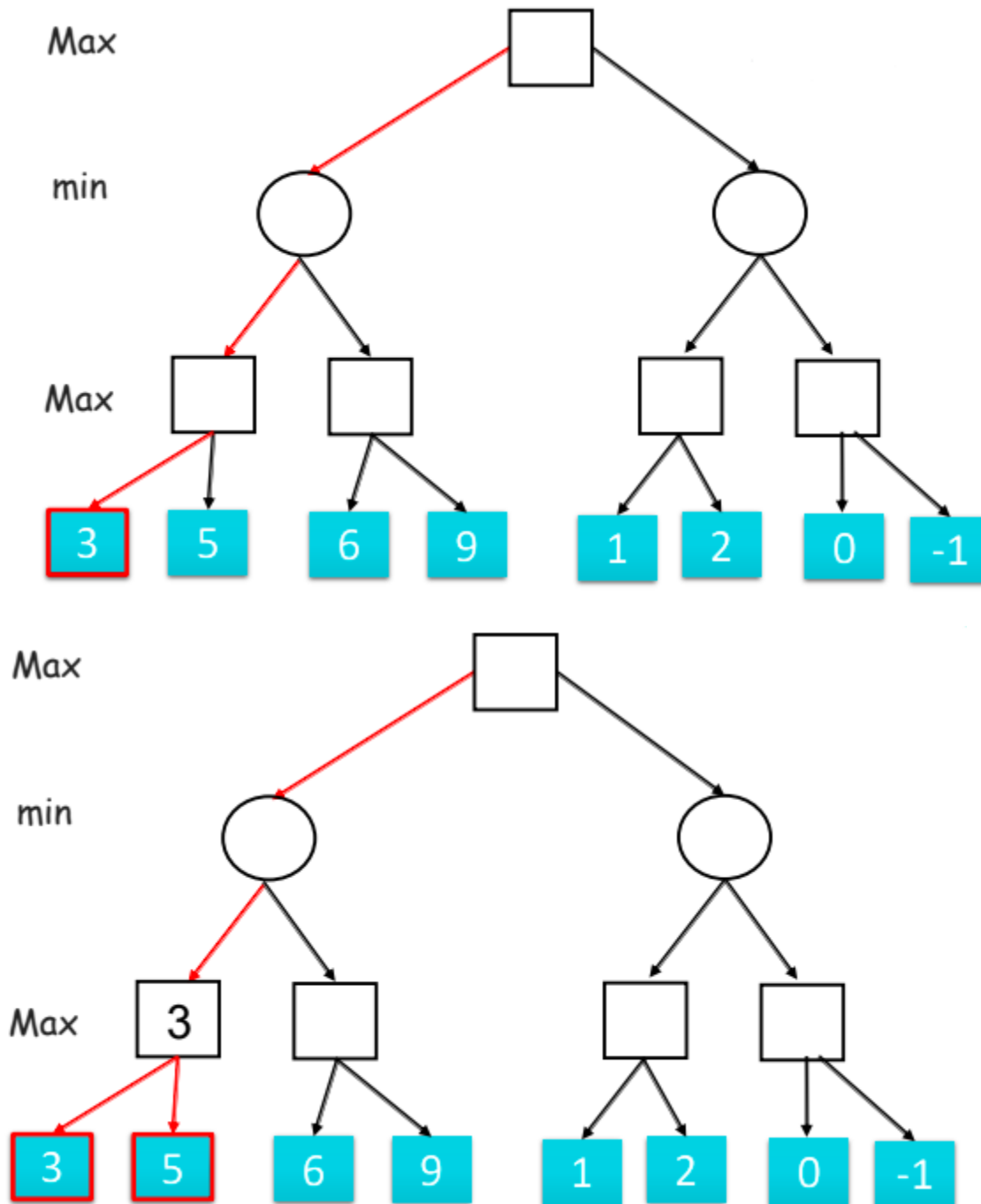
Cet arbre contient toutes les parties possibles que l'on peut jouer à partir de la position initiale. Si on a la possibilité d'explorer tout l'arbre, on peut déterminer s'il existe une stratégie gagnante pour le joueur.

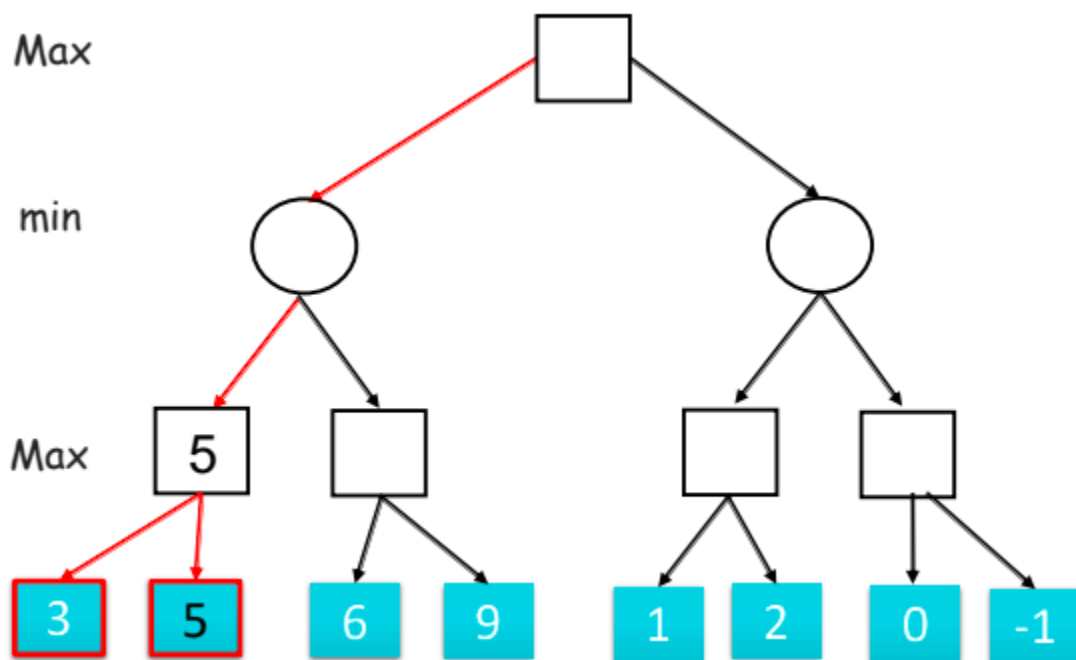
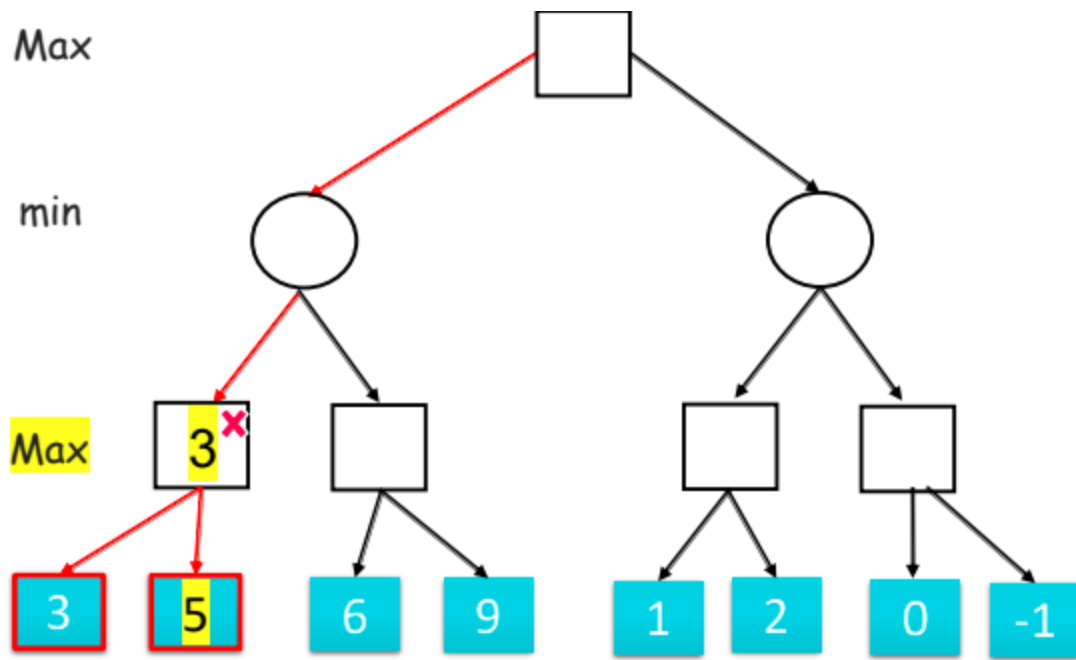
L'algorithme ***MiniMax***, dû à Von Neumann, est très simple : on visite l'arbre de jeu pour faire remonter à la racine une valeur (appelée «valeur du jeu») qui est calculée récursivement de la façon suivante :

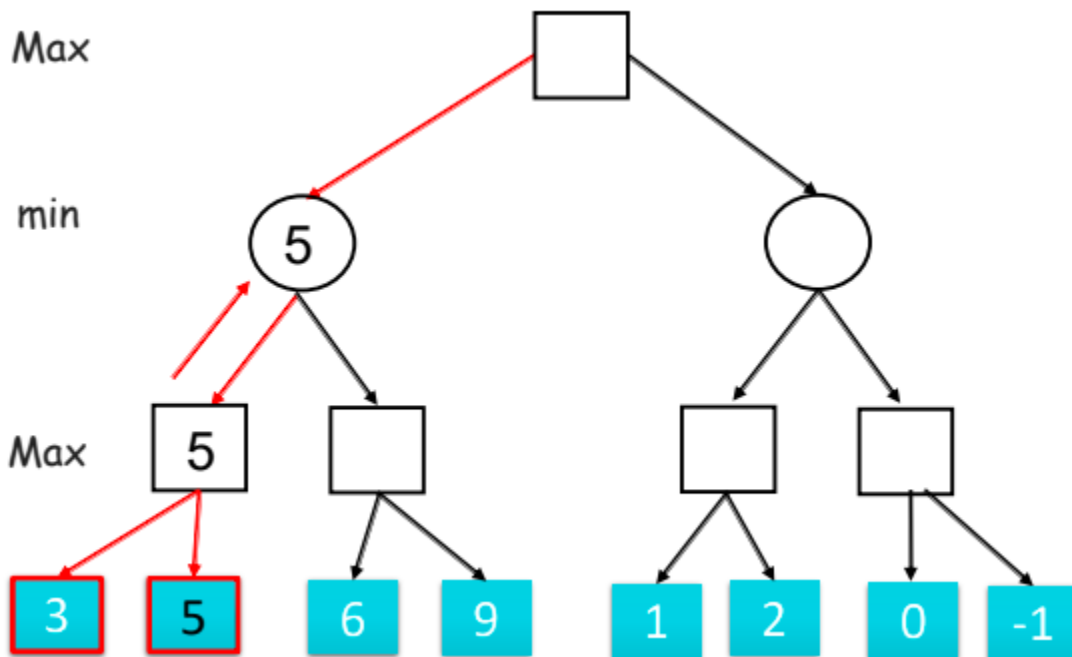
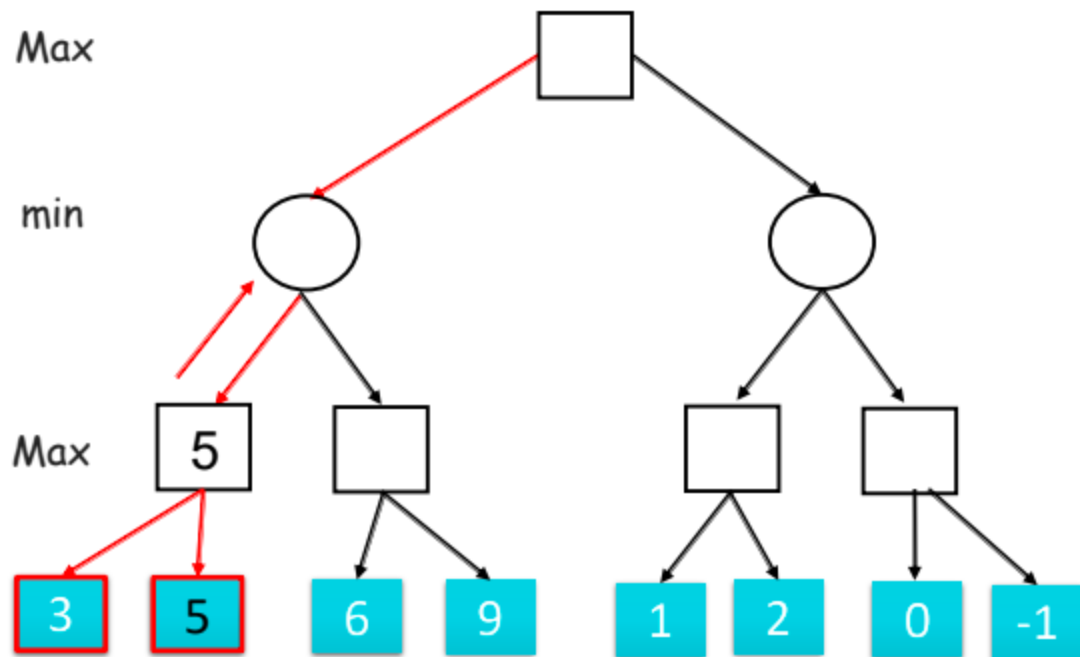
- $MiniMax(p) = h(p)$ si p est une position terminale
- $MiniMax(p) = \max(MiniMax(o_1), \dots, MiniMax(o_n))$ si p est une position joueur avec fils o_1, \dots, o_n
- $MiniMax(p) = \min(MiniMax(j_1), \dots, MiniMax(j_m))$ si p est une position opposant avec fils j_1, \dots, j_m

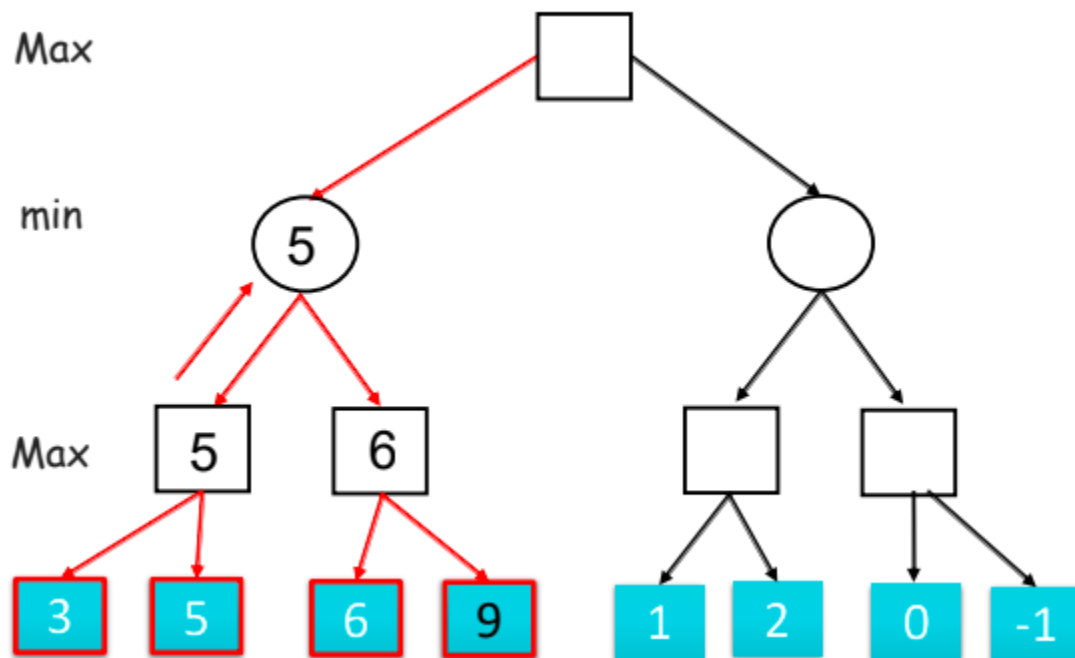
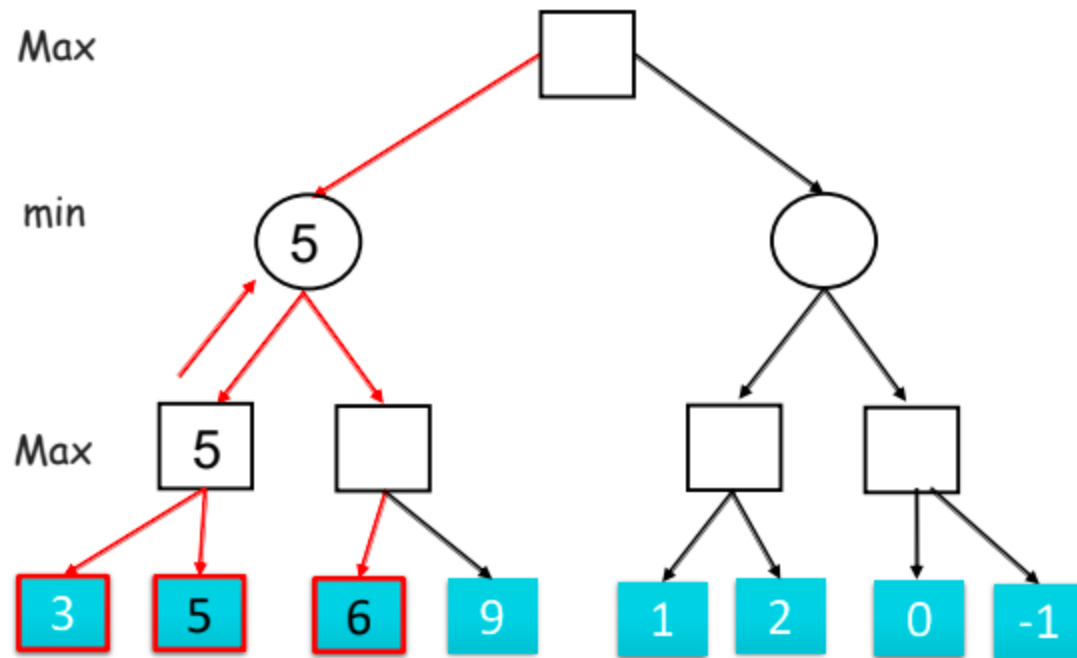
Objectif: Trouver le coup parfait pour un jeu déterministe à information parfaite

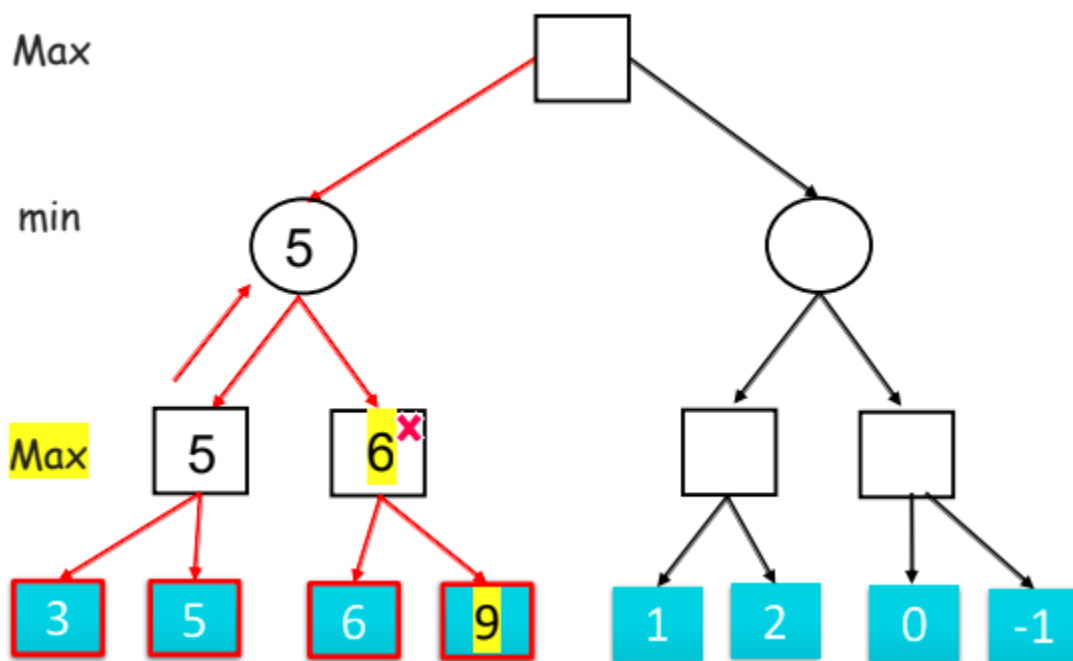
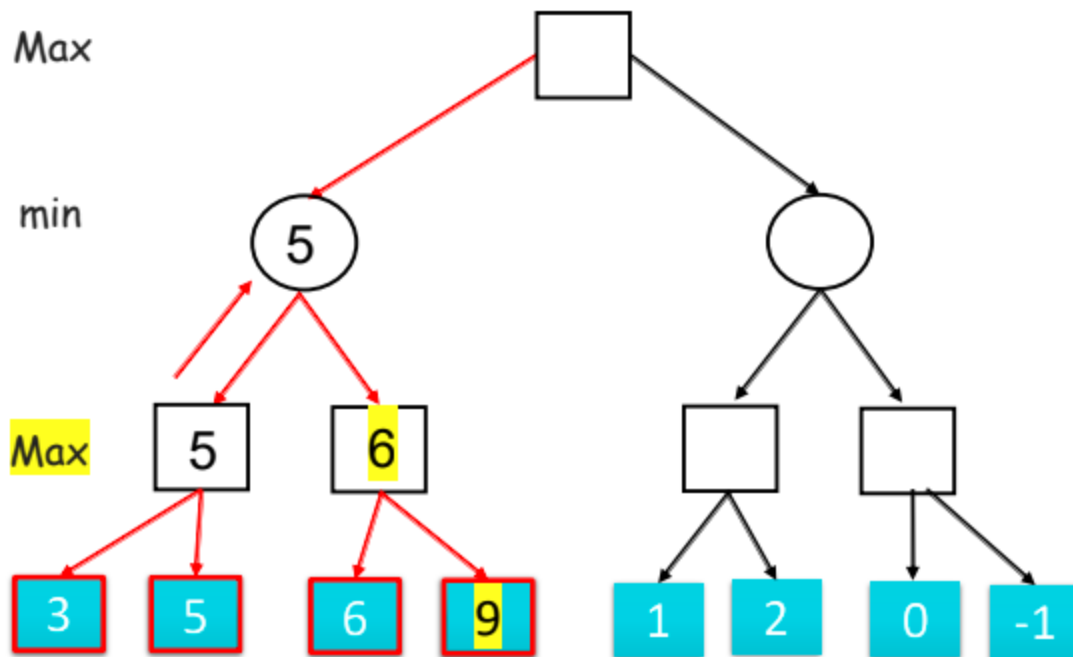
Exemple 1 :

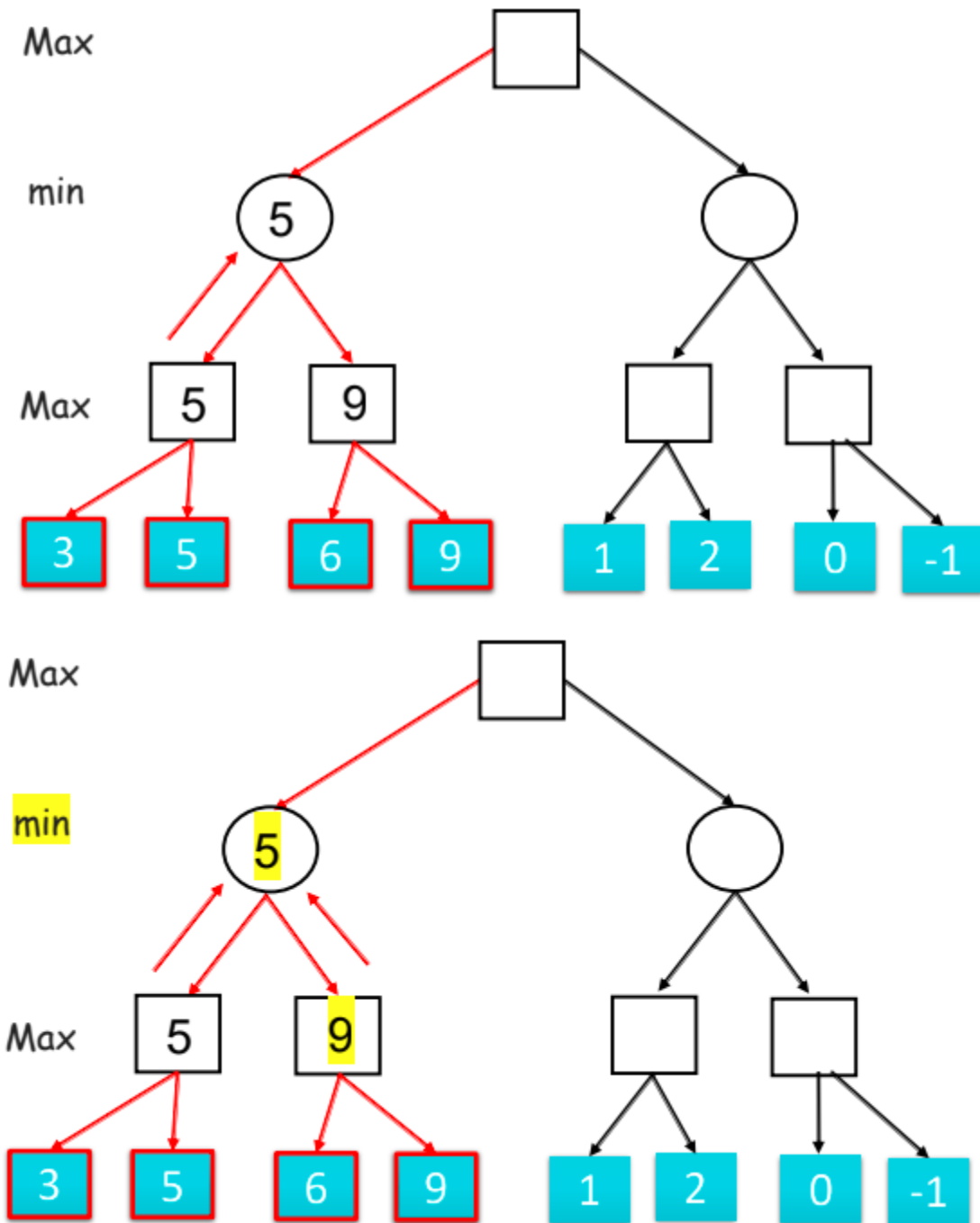


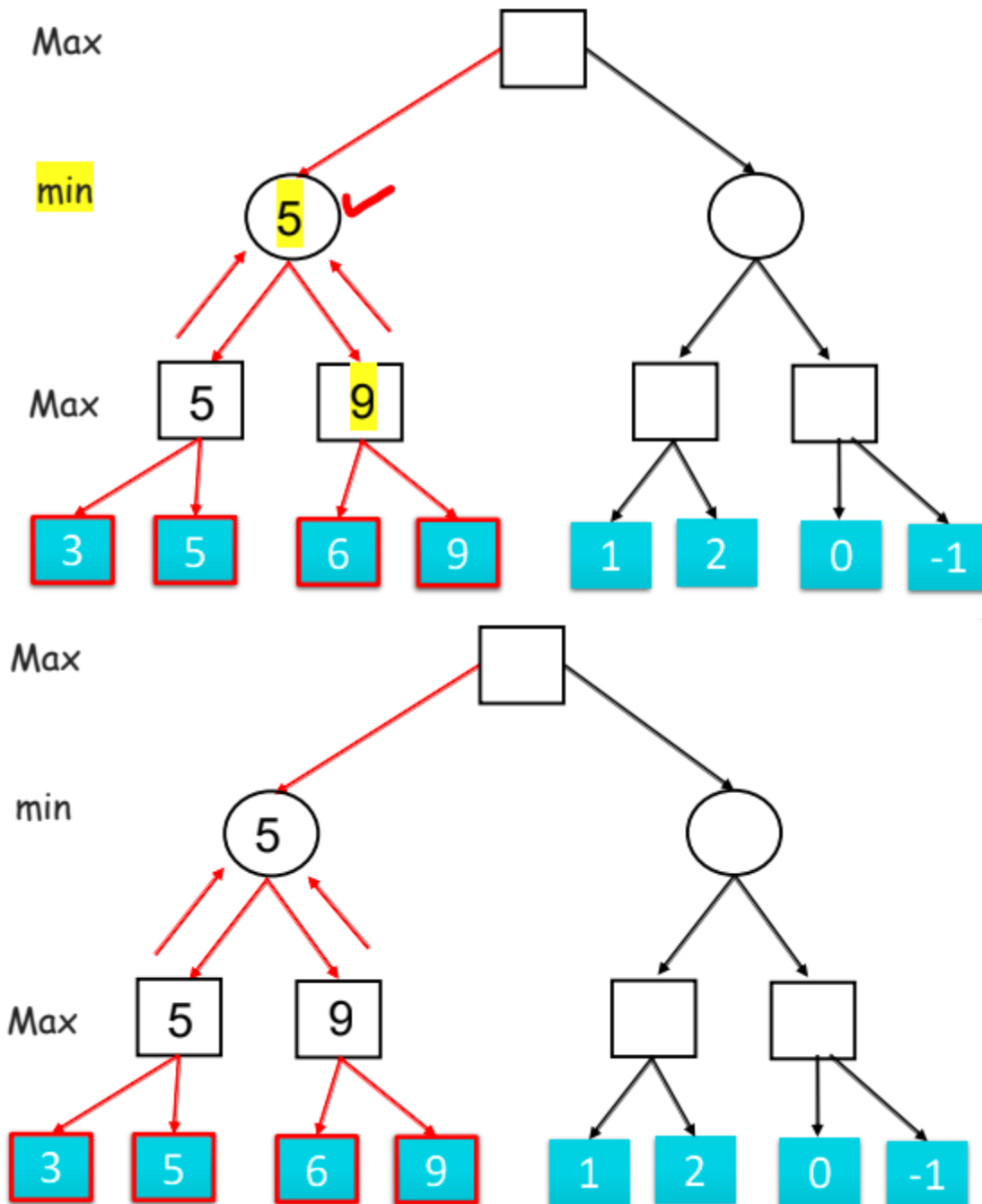


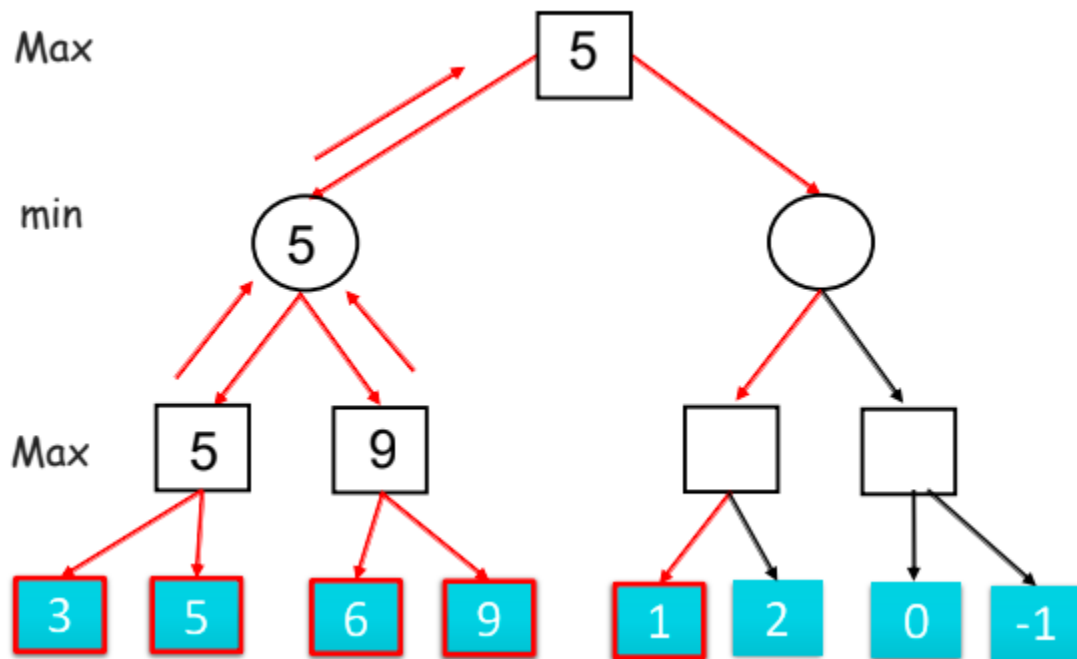
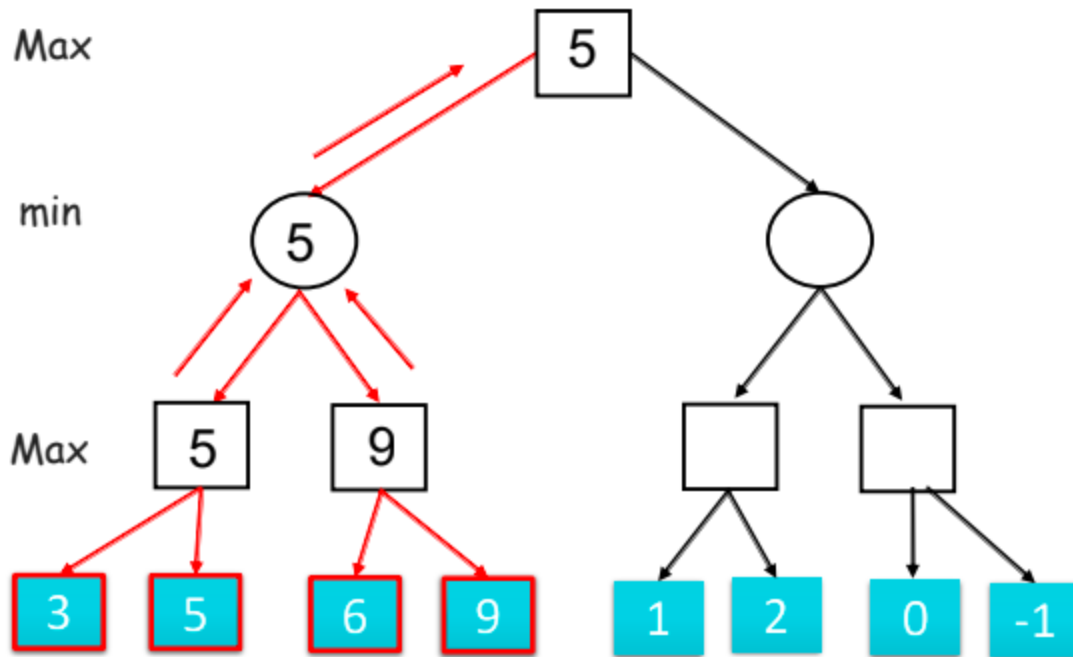


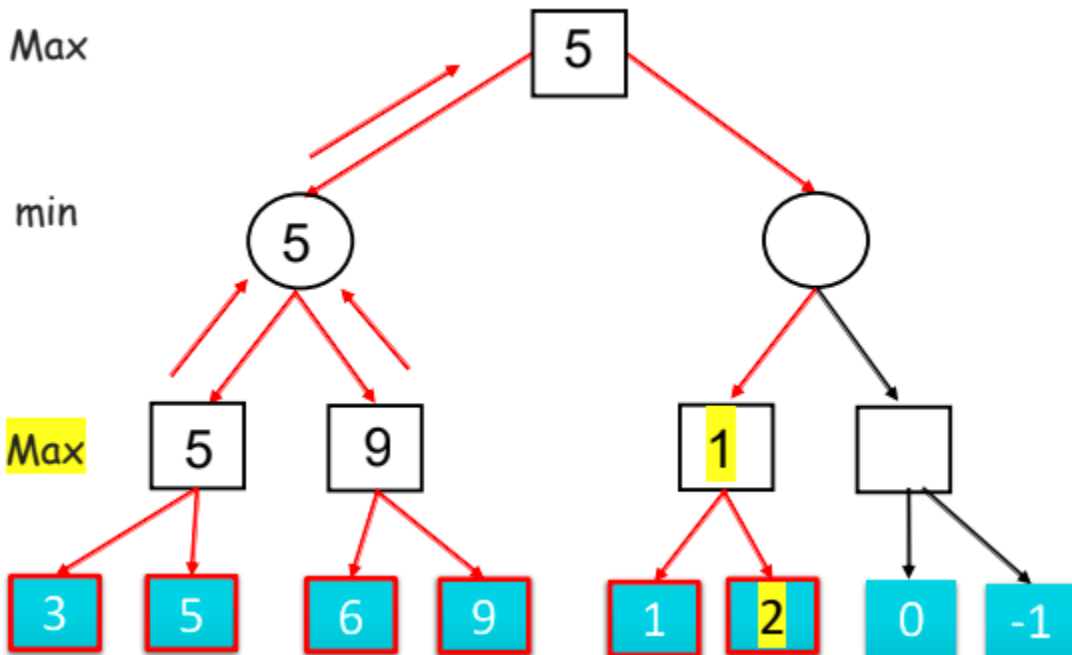
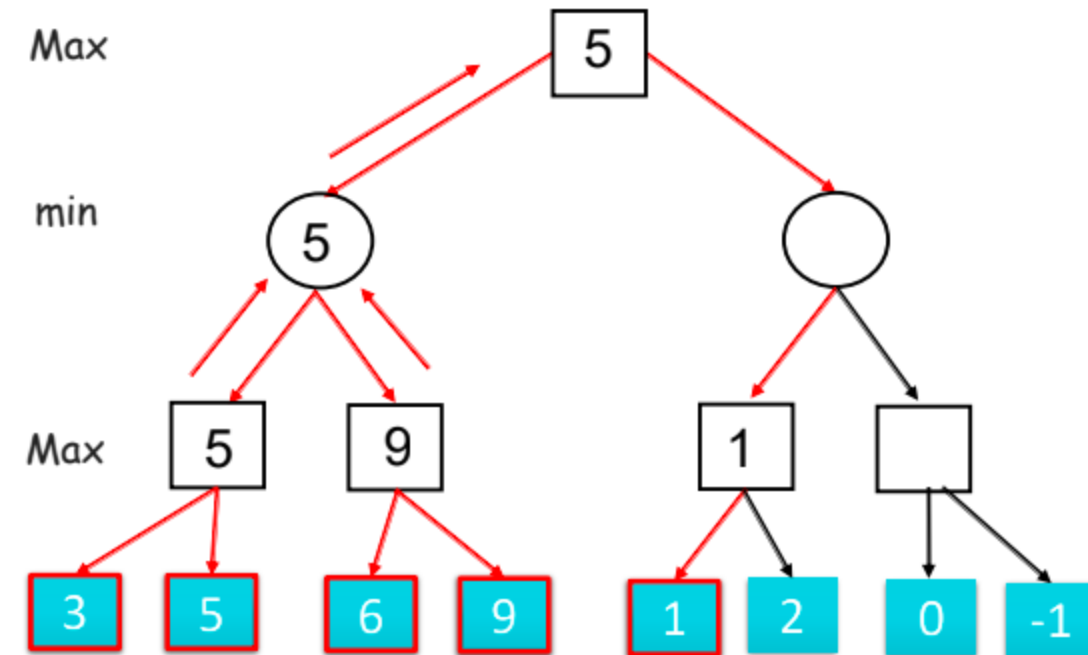


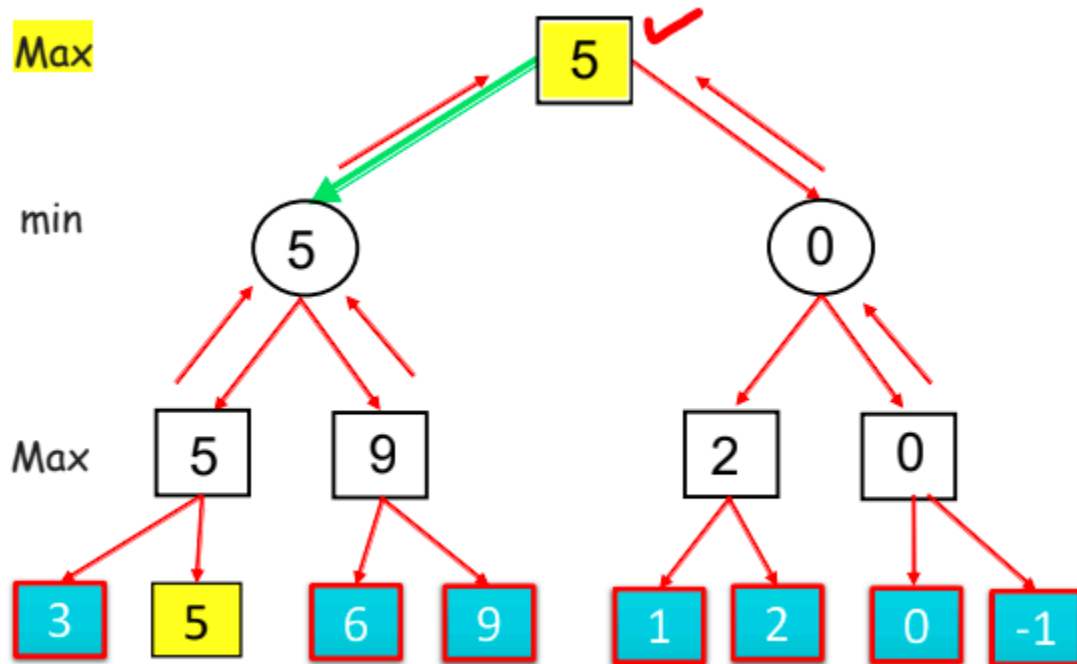
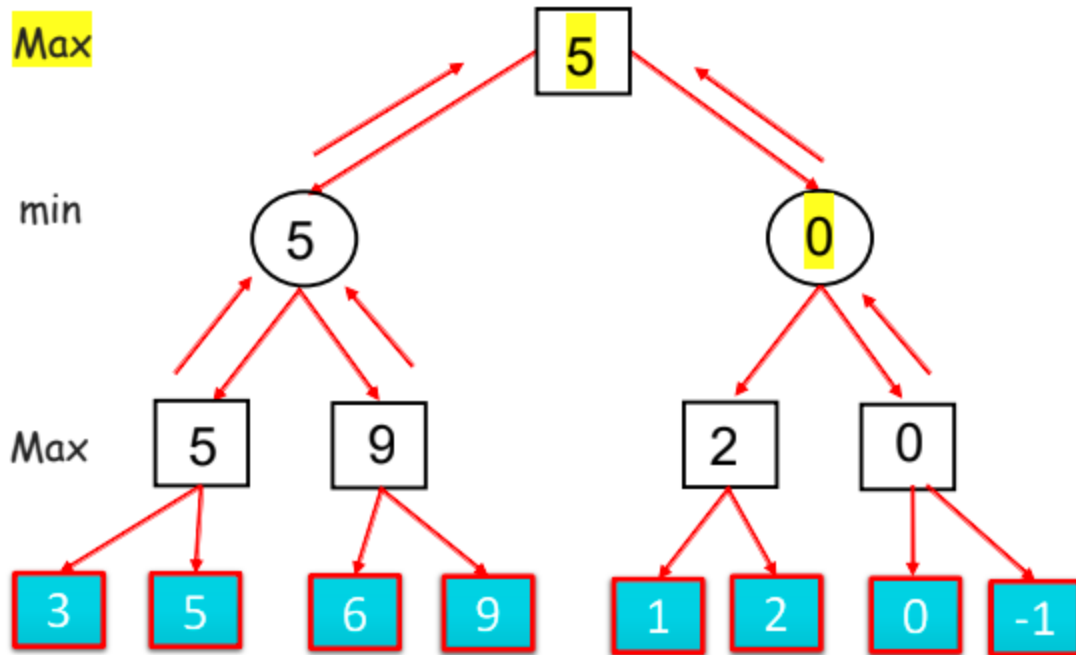












Exemple 2 :

