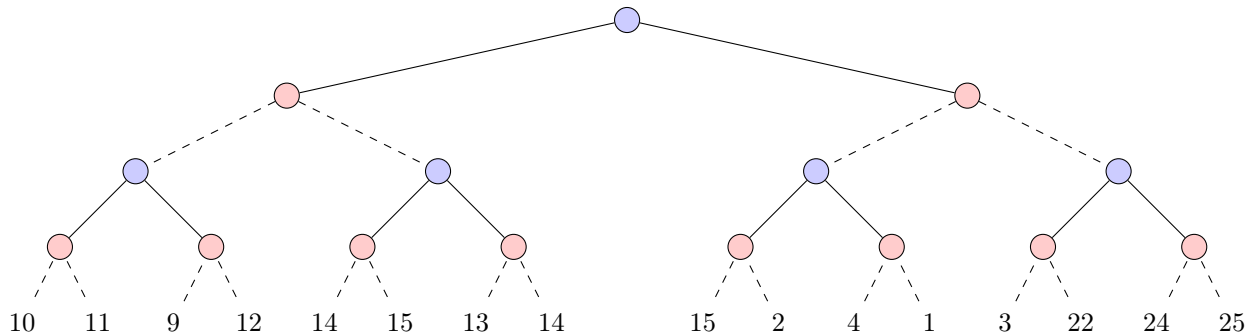


Intelligence Artificielle et Données : feuille 1

Les jeux de plateaux à deux joueurs Exploration totale, MiniMax

1 Exploration d'arbres... sur feuilles

Soit l'arbre de jeu représenté ci-dessous (les arcs venant de noeuds ennemis sont en pointillés, les valeurs aux feuilles sont les estimations heuristiques du point de vue ami) :



Quelle est la valeur MiniMax de l'arbre ?

2 Stratégies gagnantes au Morpion ?

Dans cet exercice, nous allons battre le WOPR, le super ordinateur du film Wargames, comme vu en cours (sur le Tic-Tac-Toe), autrement appelé Morpion.

Préparation : Récupérer l'archive `tictactoe.tgz` sur le site du cours.

Cette archive contient deux fichiers en Python. L'un (`Tictactoe.py`) décrit les règles du Morpion grâce à la classe `Board`. On pourrait bien entendu coder les règles de manière plus compacte, mais cette classe représente des méthodes que l'on retrouvera plus tard dans une librairie de jeu des échecs, sur laquelle on pourra quasi directement travailler avec les mêmes algorithmes de recherche. L'autre fichier est `starter-tictactoe.py` qui contient un exemple simple de déroulement d'une partie aléatoire au Morpion. Vous noterez l'utilisation des deux méthodes `push()` et `pop()` pour manipuler les coups sur le plateau (cela permet de "faire" et "défaire" des coups).

1. En utilisant les méthodes `legal_moves()` et `is_game_over()`, proposer une méthode permettant d'explorer toutes les parties possibles au Morpion (lorsque X commence). Combien y-a-t-il de parties ? Combien votre arbre de recherche a-t-il de noeuds ? Combien de temps faut-il pour tout explorer ?
2. Sans utiliser à ce stade d'horizon maximal à votre recherche arborescente, recherchez s'il existe une stratégie gagnante au Morpion (pour ce faire, vous n'aurez à prendre en compte que l'information "Gagné" / "Perdu" / "Egalité obtenue à chaque fin de partie.")
3. Améliorez votre recherche de stratégie gagnante en effectuant votre première recherche "intelligente", qui sera capable d'élaguer des parties de l'arbre de recherche tout en conservant l'admissibilité de votre recherche. Comparez les temps et le nombre de noeuds nécessaires à la recherche entre les deux approches.