

I. Jeu de solitaire... à deux joueurs

On considère un jeu dont le déroulement ressemble au jeu de solitaire, sur un plateau 3×3 . Quand deux pièces sont adjacentes, on peut sauter une pièce pour atteindre une case libre et retirer le pion par-dessus lequel on a sauté.

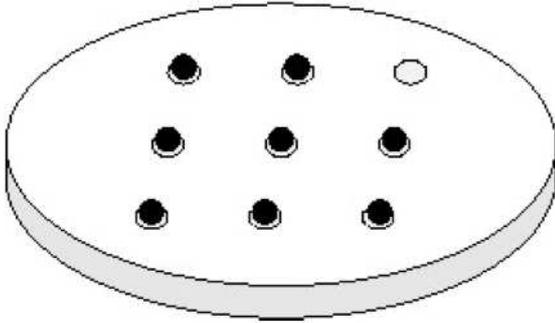
Il est obligatoire de manger un pion si cela est possible.

Les joueurs jouent à tour de rôle jusqu'à être dans l'impossibilité de sauter par-dessus un pion.

Le joueur A (Max) démarre et cherche à obtenir le plus grand nombre de pions retirés du jeu en fin de partie.

Le joueur B (Min) joue et cherche à obtenir le plus faible nombre de pions retirés du jeu en fin de partie.

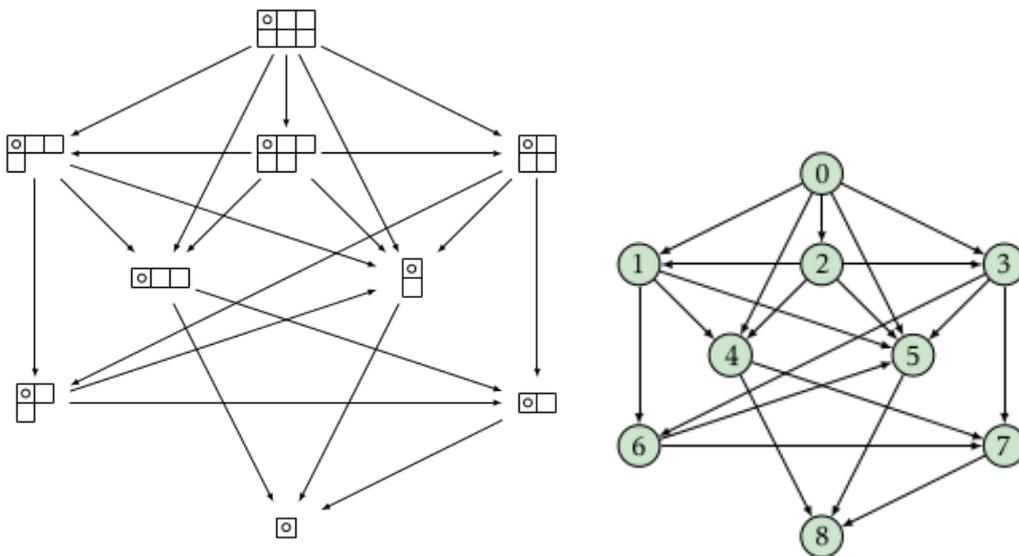
La partie est gagnée par A lorsqu'en fin de partie, 4 pions ou plus ont été mangés ; elle est gagnante pour B sinon.



1. Mettez-vous en binôme et jouez une partie.
2. Echangez les rôles et jouez une revanche.
3. Tracez l'arbre max-min de la partie et déduisez-en la meilleure stratégie pour A , et le score prévisible si B joue au mieux.

II. Calcul du noyau de champ

On reprend le jeu de champ 2×3 , à deux joueurs avec un carré empoisonné en position $(1, 1)$.



On peut représenter le graphe par le dictionnaire des successeurs.

$d = \{0: [1, 2, 3, 4, 5], 1: [4, 5, 6], 2: [1, 3, 4, 5], 3: [5, 6, 7], 4: [7, 8], 5: [8], 6: [5, 7], 7: [8], 8: [], \}$

On appelle noyau N du graphe la partie de ses sommets, telle que :

- les sommets de N n'ont aucun arc les joignant deux à deux
- chaque sommet qui n'est pas dans N a un successeur (au moins) dans N

En pratique, pour construire le noyau d'un graphe :

- On prend un cul-de-sac et ses prédécesseurs, on met l'un dans le noyau et les autres hors d'un noyau, puis on les enlève avec les arcs qui leur sont adjacents.
- On recommence avec le reste du graphe, autant de fois qu'il faut.

1. Pourquoi est-on sûr de terminer ?
2. Rédiger une fonction `sansSuccesseur(d)` qui renvoie un sommet sans successeur.
3. Rédiger une fonction `predecesseurs(d, j)` qui renvoie la liste de tous les prédécesseurs du sommet j .
4. Rédiger une fonction `supprimeSommet(d, j)` qui supprime un sommet du graphe.
5. En déduire une fonction `noyau(d)` qui renvoie la liste des sommets constituant le noyau de (S, A) .

intérêt : Si le premier joueur démarre à partir d'une situation hors du noyau, il peut s'y placer à l'issue du premier coup. Son adversaire, quoi qu'il fasse, ne peut pas aller dans le noyau au cours du 2ème coup. A son tour, le premier joueur peut se replacer dans le noyau au troisième coup. Et ainsi de suite. En allant dans le noyau à chaque fois qu'il joue, le premier joueur est sûr d'arriver à la position finale du jeu, correspondant à la victoire.

III. Jeu de morpion contre IA, stratégie par min-max

Vous pouvez télécharger le premier programme de morpion `TP6morpioneleve.py` contre une IA optimisée par minmax :

<https://capytale2.ac-paris.fr/web/c/4012-1144800>

1. Parcourez le code.
2. Pourquoi initialiser la matrice `arene` avec des zéros ?
3. Exécuter une partie contre l'IA
4. Pour chaque fonction présente dans le code, expliquez les Entrées, les sorties, et à quoi elle sert
5. Le joueur humain peut-il espérer gagner ?
6. Le joueur humain peut-il espérer ne pas perdre ?

IV. Jeu de puissance 4, stratégie par min-max

	3	4	5	5	4	3
	4	6	8	8	6	4
On peut choisir une heuristique favorisant les cases centrales	3	6	9	9	6	3
	4	6	8	8	6	4
	3	4	5	5	4	3

Vous pouvez télécharger le second programme `TP6p4eleve.py` pour mettre en oeuvre une stratégie minmax :

<https://capytale2.ac-paris.fr/web/c/4012-1144800>

1. Exécuter le code simulant une IA minimax avec heuristique et profondeur 4 contre une IA aléatoire.

```
print(Partie(s1=MiniMax4,s2=HasardTo,Print=False))
#print(Partie(s1=HasardTo,s2=HasardTo,Print=False))
#print(Partie(s1=MiniMax4,s2=HasardTo,Print=True))
```

2. Exécuter ensuite la version graphque, en décommentant

```
#print(Partie(s1=MiniMax4,s2=HasardTo,Print=True))
```

3. Enfin

```
#print(Partie(s1=HasardTo,s2=HasardTo,Print=False))
```

4. Expliquer la sortie (T, L) de la fonction `CreerGrille`, que représente T ? que représente L ?
5. Expliquer la sortie de `Jouer(T,L,joueur,col)`, lorsque le joueur joue dans la colonne `col`