1. Jeu de Nim

On considère, comme dans le cours, le jeu de Nim à un seul tas. On suppose donc disposer d'un tas de n jetons avec lequel on va jouer en enlevant au plus $k \leqslant n$ jetons à chaque fois.

1. Écrire une fonction `Nim(n:int,k:int) -> dict` qui prend en arguments les entiers `n` et `k` introduits précédemment et qui renvoie le dictionnaire représentant le graphe du jeu comme introduit dans le paragraphe **I.2** du cours ; on ne considèrera pas le sommet `0` (comme vu dans la suite du cours).

2. Écrire une fonction `biparti(G:dict) -> dict` qui prend en argument le dictionnaire d'un graphe comme construit juste avant et qui renvoie le dictionnaire du graphe biparti associé. Les sommets de ce graphe seront représentés par des couples `(i,j)` avec `j=0` pour les positions contrôlées par le joueur J_1 (et `i` le numéro du sommet) et avec `i=0` pour les positions contrôlées par le joueur J_2.

Par exemple le graphe biparti associé au jeu de Nim avec $n = 5$ et $k = 2$ est

`{(1,0):[] , (0,1):[] , (2,0):[(0,1)] , (0,2):[(1,0)] , (3,0):[(0,1),(0,2)] , (0,3):[(1,0),(2,0)] , (4,0):[(0,2),(0,3)] , (0,4):[(2,0),(3,0)] , (5,0):[(0,3),(0,4)] , (0,5):[(3,0),(4,0)]}`

3. L'attracteur du joueur J_1 (qui cherche à atteindre la position finale `(0,1)`) est, dans les cas $(n, k) = (5, 2)$, $(n, k) = (7, 2)$ et $(n, k) = (20, 3)$:

a) `[(0,1),(2,0),(3,0),(0,4),(5,0)]` pour $(n, k) = (5, 2)$

b) `[(0,1),(2,0),(3,0),(0,4),(5,0),(6,0),(0,7)]` pour $(n, k) = (7, 2)$

c) `[(0,1),(2,0),(3,0),(4,0),(0,5),(6,0),(7,0),(8,0),(0,9),(10,0),(11,0),(12,0),(0,13),(14,0),(15,0),(16,0),(0,17),(18,0),(19,0),(20,0)]` pour $(n, k) = (20, 3)$

Dans chacun des trois cas, si le joueur J_1 joue en premier, quel joueur possède une stratégie gagnante ?

Si J_1 possède une stratégie gagnante, combien de jetons doit-il prendre lors de son premier coup ?

4. On suppose que le joueur J_1 possède une stratégie gagnante depuis la position `(n,0)`. Écrire une fonction `jeu(G:dict,n:int) -> dict` qui renvoie un dictionnaire qui donne la stratégie à adopter pour que le joueur J_1 gagne la partie : les clés du dictionnaires sont les positions contrôlées par J_1 par lesquelles il est susceptible de passer et la valeur associée est la position sur laquelle il doit aller pour être sûr de gagner.

2. Positions finales

On considère un jeu sur un graphe biparti `G` dont les sommets contrôlés par J_1 sont de la forme `(i,0)` et ceux contrôlés par J_2 de la forme `(0,j)`. Comme dans le jeu de Nim précédent, on suppose que les positions à atteindre pour qu'un des joueur gagne sont les positions contrôlées par son adversaire dans lesquelles il ne peut plus jouer.

Écrire une fonction `Final(G:dict,k:int) -> list` qui prend en argument le dictionnaire `G` du graphe et un entier `k`$\in \{1, 2\}$ et qui renvoie la liste des positions à atteindre pour le joueur J_k.

3. Attracteur récursif

Écrire une fonction qui détermine l'attracteur d'un joueur à partir de la liste de ses positions à atteindre, comme en cours, mais qui calcule les ensembles A_i de façon récursive. On supposera que, comme dans les exercices précédents, les sommets contrôlés par J_1 sont de la forme `(i,0)` et ceux contrôlés par J_2 de la forme `(0,j)`.