TD Informatique TSI2

Algorithms of games

EXERCICE 01

Graphe biparti ou non

1. Dessiner un graphe dont les 10 sommets sont des ronds numérotés de 0 à 9, que l'on placera de façon à peu près équidistants sur un grand cercle de telle manière que le rond marqué 1 soit en haut sur le cercle et qu'on mette les ronds 2, 3, 4, 5, 6, 7, 8, 9 et 0 dans le sens inverse du sens trigonométrique.
   Les arêtes (uniques entre chaque rond) sont telles que :
   Le rond 0 est relié avec les ronds $1, 8$ et 9.
   Le rond 1 est relié avec les ronds $0, 3, 4, 7$.
   Le rond 2 est relié avec les ronds $3, 5$ et 7.
   Le rond 3 est relié avec les ronds $1, 2, 6$ et 9.
   Le rond 4 est relié avec les ronds $1, 6, 8$ et 9.
   Le rond 5 est relié avec les ronds $2, 8$ et 9.
   Le rond 6 est relié avec les ronds $3, 4$ et 7.
   Le rond 7 est relié avec les ronds $1, 2, 6$ et 8.
   Le rond 8 est relié avec les ronds $0, 4, 5$ et 7.
   Le rond 9 est relié avec les ronds $0, 3, 4$ et 5.
2. Ce graphe est-il biparti ? On pourra utiliser la méthode de coloration.
3. Définir un dictionnaire représentant ce graphe et utiliser la procédure `cycle_impair` du cours pour vérifier votre résultat.

EXERCICE 02

Jeu de Nim avec un seul tas et quatre objets

Chaque joueur retire un ou deux objets du tas à tour de rôle.
Le dernier joueur qui retire des objets a gagné la partie.

1. Définir un graphe `G` sous forme d'un dictionnaire dont chaque clé, représentant un sommet, est un couple composé du numéro du joueur (1 ou 2) qui contrôle le sommet et du nombre d'objets restant dans le tas. Les valeurs associées aux clés sont des listes de couples où chaque couple représente un sommet après la prise du joueur qui contrôle le sommet représenté par la clé.

   Le premier joueur est le joueur 1 et il y a 4 objets. Le sommet de départ est donc représenté par $(1, 4)$. Le joueur 1 peut retirer un ou deux objets et les deux sommets atteints sont contrôlés par le joueur 2. Alors `G = (1,4):[(2,3),(2,2)],...`
2. On veut retourner par une procédure le dictionnaire `G` demandé à la question **1**.
   (a) Pour cela, on commence par construire une procédure `nim(n,j,G)` récursive où `n` est le nombre de sommets (donc d'objets), `j` est le numéro d'un joueur (1 ou 2) et `G` le graphe que l'on initialise à `G = { }` ensuite. On remarque si `j` est un joueur alors `1+j%2` est l'autre joueur.
   Ainsi dans le corps de la procèdure `nim(n,j,G)` :

```
# Pour la condition  n>1 , on tape :
G[(j,n)] = [(1+j%2,n-2),(1+j%2,n-1)]
nim(n-1,1+j%2,G)
nim(n-2,1+j%2,G)
# Pour la condition  n vaut 1 , on tape :
G[(j,n)] = [(1+j%2,n-1)]
nim(n-1,1+j%2,G)
# Sinon, on tape :
G[(j,n)] = []
```

(b) Taper ensuite :

```
In [1] : def jeu(n):
            G={}
            nim(n,1,G)
            return G
```

Retrouver le graphe `G` pour $n = 4$ de la question **1**.

EXERCICE 03

Jeu de chomp

On considère le jeu de chomp avec une tablette composée de 2 lignes et 3 colonnes. Le carré $(1, 1)$ est grisé et empoisonné. Les différents états de la tablette sont numérotés.

Etat 1

■		

Etat 2

■		
		eaten

Etat 3

■		eaten
		eaten

Etat 4

■		
	eaten	eaten

Etat 5

■		
eaten	eaten	eaten

Etat 6

■		eaten
	eaten	eaten

Etat 7

■		eaten
eaten	eaten	eaten

Etat 8

■	eaten	eaten
	eaten	eaten

Etat 9

■	eaten	eaten
eaten	eaten	eaten

Pour simplifier le graphe et sauver les joueurs, on considère qu'un joueur ne peut pas manger le carré emposonné. Donc l'état 9 est un état final. Le joueur qui se trouve devant cet état a perdu la partie car il ne peut plus jouer.

1. Dessiner le graphe du jeu. On pourra faire deux colonnes. Un sommet est un couple avec pour premier élément le numéro du joueur et pour second élément létat de la tablette. Dans la première colonne, on mettra les huits sommets `(1,i)` contrôlés par J_1 les uns sous les autres en commençant par `(1,1)` et dans la deuxième colonne, on mettra les huits sommets `(2,i)` contrôlés par J_2 les uns sous les autres commençant par `(2,2)`. Puis on placera les différentes arêtes entre les deux colonnes. **Remarque** : *attention à deux notations identiques pour deux types d'objets. Le sommet `(i,j)` désigne que le joueur numéro i est à l'étape j et le carré (i, j) désigne le carré de la tablette qui est à la ligne i et colonne j.*
2. En déduire le graphe `G` du jeu sous la forme d'un dictionnaire.
3. On suppose que J_1 depuis le sommet `(1,1)` mange le carré $(2, 3)$ et atteint donc le sommet `(2,2)`. Au tour de J_2. Justifier que J_2 n'a que quatre possibilités. Et prévoir ce que fera J_1 à chaque fois pour gagner en un coup ou en deux coups suivants. Conclure que l'on a trouvé une stratégie gagnante pour J_1.