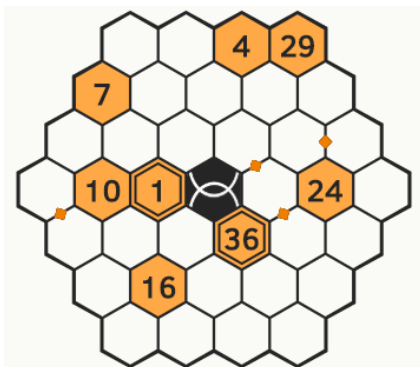


# ● Projet INF402 : Le Rikudo

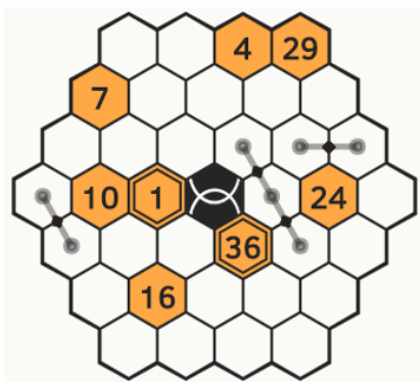
## I) Introduction : Présentation du jeu

Le but du jeu est de placer les nombres de 1 à n dans la grille (n correspondant au nombre de cases) de façon à former un chemin de nombres consécutifs. Deux nombres consécutifs doivent donc être voisins. Un lien entre deux cases indique un point de passage du chemin. À la fin, toute la grille doit être remplie. En général, la première et la dernière case sont données dès le départ.

Exemple :



On remplit la grille fournie (ici à gauche) avec les nombres de 1 à 36.



Les liens entre les cases réduisent les possibilités de résolution.



La grille résolue permet de tracer un chemin de nombres consécutifs de 1 à 36.

## II) Modélisation des règles du jeu sous forme de symboles logiques

On note  $C_i^j$  la variable booléenne “La case  $i$  contient le nombre  $j$ ”.

### Règles générales

- Dans chaque case, il y a un nombre :  
$$\forall i \in [1; n], \exists j \in [1; n], C_i^j$$
- Chaque nombre se trouve dans une case :  
$$\forall j \in [1; n], \exists i \in [1; n], C_i^j$$
- Une case ne contient qu'un seul nombre :  
$$\forall i \in [1; n], \forall j \in [1; n], \forall k \in [1; n], C_i^j \Rightarrow k \neq j \Rightarrow \neg C_i^k$$
- Un nombre ne peut se trouver que dans une seule case :  
$$\forall i \in [1; n], \forall j \in [1; n], \forall l \in [1; n], C_i^j \Rightarrow i \neq l \Rightarrow \neg C_l^j$$
- Toutes les cases ont un voisin direct dont le nombre est le suivant, sauf la dernière :  
$$\forall i \in [1; n], C_i^n \vee (\exists j \in [1; n], \text{voisin}(i, j) \wedge (\exists k \in [1; n - 1], C_i^k \wedge C_j^{k+1}))$$

La variable  $\text{voisin}(i, j)$  est “Les cases  $i$  et  $j$  sont voisines”. Pour avoir moins de variables on ne créera qu'une seule variable  $v(i, j)$  où  $i \leq j$  et on a  $\text{voisin}(i, j) = \text{voisin}(j, i) = v(i, j)$ .

### Règles spécifiques à une configuration

- On ajoute pour chaque pont entre  $i$  et  $j$  deux cases:  
$$\exists k \in [1; n - 1], (C_i^k \wedge C_j^{k+1}) \vee (C_i^{k+1} \wedge C_j^k)$$
- On ajoute  $\text{voisin}(i, j)$  ou  $\neg \text{voisin}(i, j)$  selon si les cases sont voisines ou non
- On ajoute pour les cases  $i$  préremplies avec  $j$   $C_i^j$ .

## Format des fichiers

### Le fichier décrivant une grille à résoudre

Voici le format du fichier:

n (nombre de cases)

nb arêtes

(**\***|**#**)[0-9]+ [0-9]+ Pont Pas pont Case 1 Case 2 (Case 1 < Case 2)

nb cases données

[0-9]+ [0-9]+ Case Valeur

Exemple

36

98