**Descrição da Solução:**

Lê-se o input to stdin, traduzindo-o para uma instância de *Board (board)*, pela função parse\_instance\_from\_stdin. O *board* é uma representação de um problema takuzu.

Um *Board* tem os atributos:  
- *size*, inteiro que representa o número de linhas e colunas;   
- *rows*, uma lista de listas, e a sua transposta, *cols*, tal que *rows*[i][j] e *cols*[j][i] representam ambos o número da linha i, coluna j (*cols* foi criada para facilitar verificações na vertical); *maxNrsPerLine*, inteiro que corresponde ao máximo número de 0’s ou 1’s que podem estar em cada linha (*ceiling* de *size*/2);   
- *nrMissing*, inteiro que corresponde ao número de posições por preencher;   
- *rowStatus*, dicionário que contém o estado atual das linhas de *rows* guardado em 3 entradas na forma de listas: “*Zeroes”* (lista do número de 0’s por linha), *“Ones”* (lista do número de 1’s por linha), e *“Missing”* (lista do número de posições por preencher por linha);  
- *colStatus*, dicionário semelhante a *rowStatus* mas para as colunas de *cols.*

Após a geração de um *TakuzuProblem* *takuzu* a partir deste *board*, efetua-se uma procura em profundidade (*depth\_first\_tree\_search(takuzu)*) com o objetivo de encontrar a solução do takuzu. Para tal, para cada estado, calcula-se quais as ações possíveis ( *actions()* ). A partir de um certo estado, existem ações possíveis (de acordo com o *board* do estado atual, é possível colocar o valor v na posição [i][j] sem quebrar imediatamente as restrições do problema), ações impossíveis, e ações certas (de acordo com o *board* do estado atual, para uma dada posição, apenas é possível colocar o valor v na posição [i][j] sem quebrar imediatamente as restrições do problema). Em *actions()*, procura-se devolver uma única ação certa assim que esta é encontrada – e, à falta de uma, apenas duas ações possíveis mas não certas para uma mesma posição. Tudo é efetuado, sempre, pela mesma ordem:

1. Ações certas (relembrar que, se se encontrar uma ação numa sub-secção, devolve-se essa e apenas essa imediatamente, não se passando nem pelas sub-secções seguintes nem pela secção 2.):
   1. Para cada linha não preenchida: se o número de 0’s ou 1’s é o máximo permitido, a ação é preencher a primeira posição vazia da linha com o valor que não atingiu lotação máxima
   2. Para cada coluna não preenchida: se o número de 0’s ou 1’s é o máximo permitido, se o número de 0’s ou 1’s é o máximo permitido, a ação é preencher a primeira posição vazia da coluna com o valor que não atingiu lotação máxima;
   3. Para cada posição do *board*, se o conjunto dos valores dela própria mais as suas posições imediatamente adjacentes na horizontal tem dois valores repetidos (e o terceiro é nulo), a ação é colocar o valor não-presente na posição vazia
   4. Para cada posição do *board*, se o conjunto dos valores dela própria mais as suas posições imediatamente adjacentes na vertical tem dois valores repetidos (e o terceiro é nulo), a ação é colocar o valor não-presente na posição vazia
2. Ações possíveis, mas não certas (da esquerda para a direita, de cima para baixo; relembrar que, para entrar nesta secção, é necessário que não existam ações certas):
   1. As duas ações são: preencher a primeira posição vazia com 0, preencher a primeira posição vazia com 1

**Heurística**

A heurística utilizada devolve 1 para todos os nodes; como