TPC2 - MFES Leonardo de Freitas Marreiros pg47398 MEI

1. Futoshiki Puzzle

Futoshiki é um puzzle lógico japonês jogado num tabuleiro N × N, onde são assinaladas restrições de desigualdade entre entre algumas posições contiguas do tabuleiro.

O objetivo é colocar os números 1..N de forma a que cada número não apareça repetido em cada linha nem em cada coluna do tabuleiro, e que as relações de desigualdade assinaladas sejam respeitadas. Alguns números podem estar fixos no tabuleiro inicial. Pode ver mais informações sobre o puzzle em:

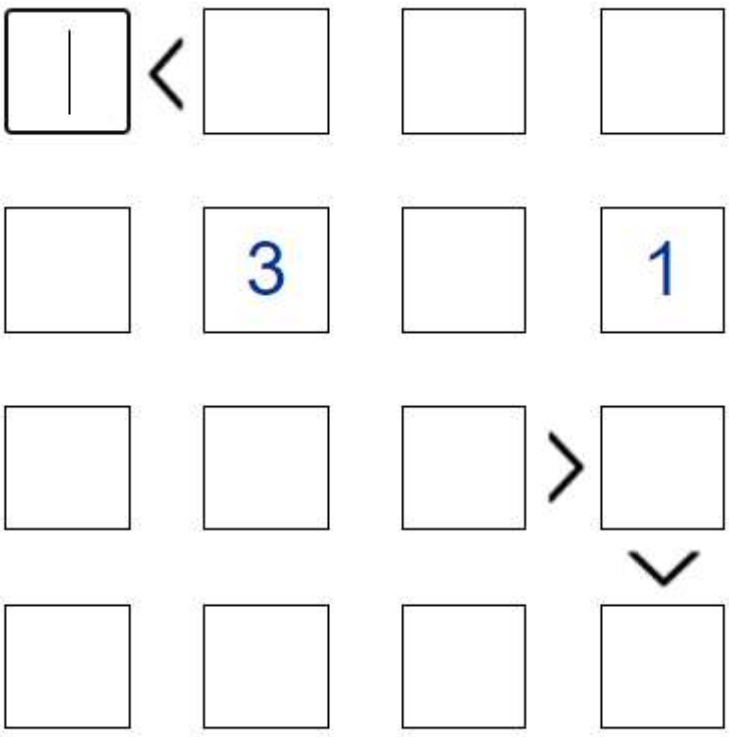
- http://en.wikipedia.org/wiki/Futoshiki
- http://www.brainbashers.com/futoshiki.asp

Desenvolva um programa em Phyton para resolver este jogo como auxílio de um SMT solver.

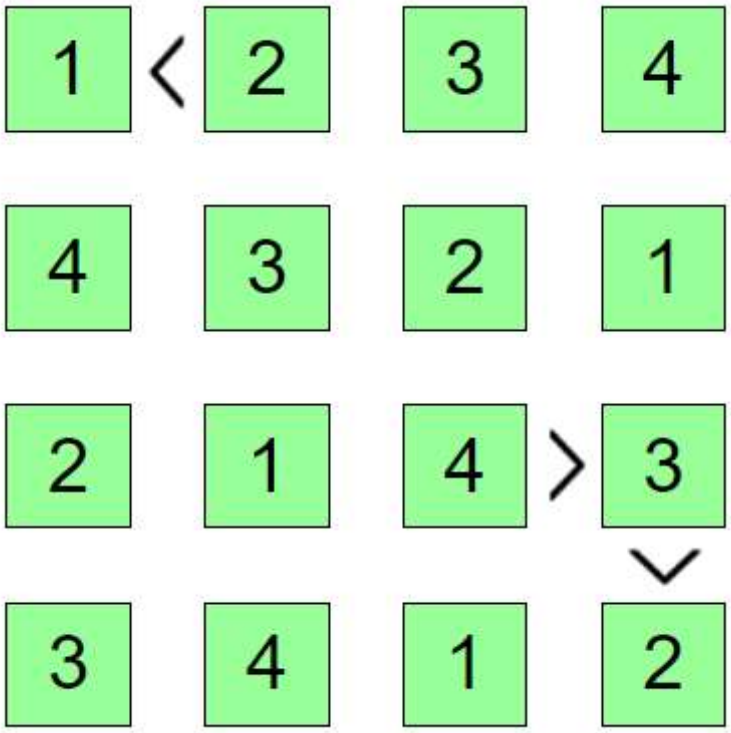
- **Input:** a configuração do tabuleiro inicial deverá ser fornecida num ficheiro de texto, em formato que entendam adquado para o descrever.
- **Output:** a solução do puzzle deverá ser impressa no ecrã.

2. Tabuleiros - Exemplo

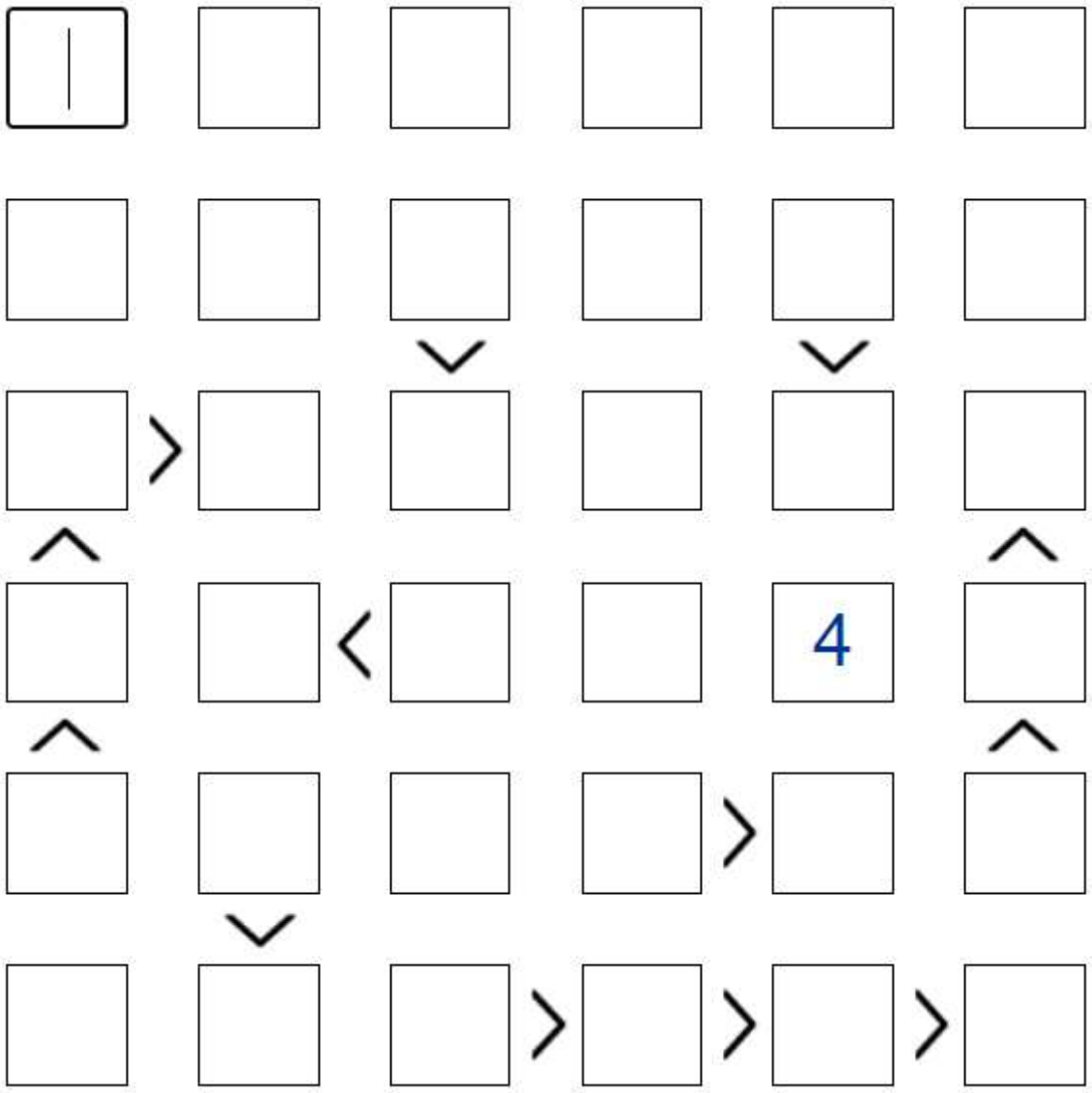
Input 1



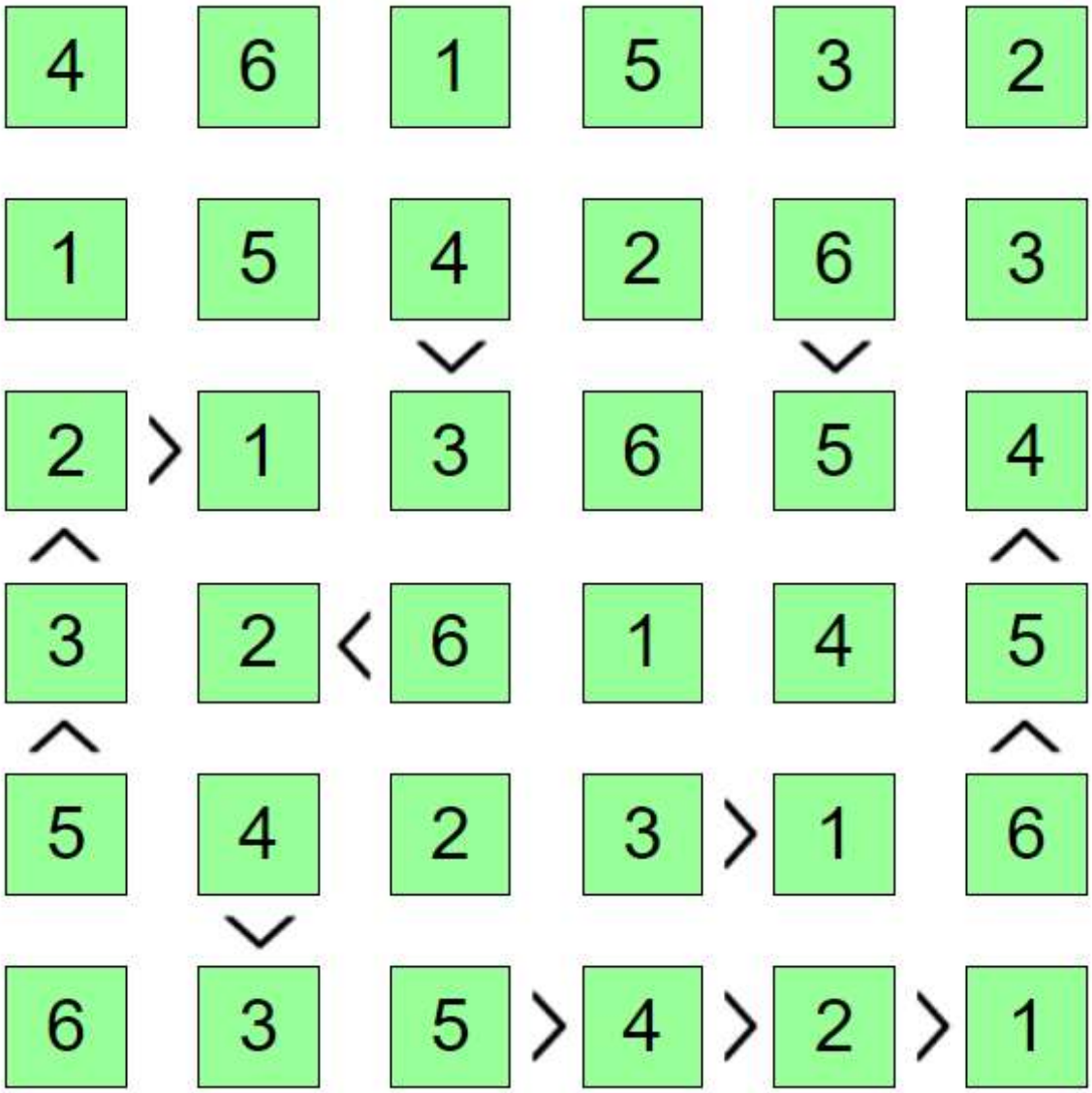
Output 1



Input2



Output 2



3. Estrutura do Tabuleiro

Exemplo

```
4

_ < _ = _ = _ end
- - - - mid
_ = 3 = _ = 1 end
- - - - mid
_ = _ = _ > _ end
- - - \ mid
_ = _ = _ = _ end
```

A primeira linha do ficheiro corresponde à dimensão do tabuleiro. As restantes linhas correspondem à tradução do tabuleiro: linhas que terminam com **end** correspondem às linhas do tabuleiro, linhas que terminam com **mid** correspondem à informação entre as linhas.

Legenda:

- **_** : célula por preencher;
- **=** : não existe qualquer restrição entre valores adjacentes na mesma linha;
- **>** : uma célula vizinha tem de ser maior que a outra na mesma linha
- **<** : uma célula vizinha tem de ser menor que a outra na mesma linha
- **-** : células sem restrição entre o valor superior e inferior
- **/** : célula superior tem de ser menor que a célula inferior
- **** : célula superior tem de ser maior que a célula inferior

```
!pip install z3-solver
```

```
Collecting z3-solver
  Downloading z3_solver-4.8.12.0-py2.py3-none-manylinux1_x86_64.whl (33.0 MB)
     |████████████████████████████████| 33.0 MB 18 kB/s
Installing collected packages: z3-solver
Successfully installed z3-solver-4.8.12.0
```

```
from z3 import *
```

4. Solução

```
#Tabuleiro 1
'''
4

_ < _ = _ = _ end
- - - - mid
_ = 3 = _ = 1 end
- - - - mid
_ = _ = _ > _ end
- - - \ mid
_ = _ = _ = _ end

'''

#Tabuleiro 2
'''
6

_ = _ = _ = _ = _ = _ end
- - - - - - mid
_ = _ = _ = _ = _ = _ end
- - \ - \ - mid
_ > _ = _ = _ = _ = _ end
/ - - - - / mid
_ = _ < _ = _ = 4 = _ end
/ - - - - / mid
```

```
    _ = _ = _ = _ > _ = _ end
    - \ - - - - mid
    _ = _ = _ > _ > _ > _ end
    '''


    f = open("tabuleiro", "r")


    N = int(f.readline())

    s = Solver()

    x = {}
    for i in range(N):
        x[i] = {}
        for j in range(N):
            x[i][j] = Int('x'+str(i)+str(j))       # declaração de variáveis
            s.add(And(1<= x[i][j], x[i][j]<=N))    # restrições de valor

    # restrições de linha
    for i in range (N):
      s.add(Distinct([x[i][j] for j in range(N)]))
    # restrições de coluna
    for j in range (N):
      s.add(Distinct([x[i][j] for i in range(N)]))

    s.push()

    l = 0
    c = 0
    for line in f:
      tok = line.split()
      for t in tok:
        if t == 'end':
          l += 1
          c = 0
        elif t == 'mid':
          c = 0
        elif t == '<':
          s.add(x[l][c-1] < x[l][c])
        elif t == '>':
          s.add(x[l][c-1] > x[l][c])
        elif t == '=':
          continue
        elif t == '_' or t == '-':
          c += 1
        elif t == '/':
          s.add(x[l-1][c] < x[l][c])
          c += 1
        elif t == '\\':
          s.add(x[l-1][c] > x[l][c])
          c += 1
        else:
          s.add(x[l][c] == int(t))
```

```
        c += 1

r = s.check()
if r==sat :
  m = s.model()
  #print(m)
  #print("")
  print('== SOLUÇÃO ==')
  for i in range (N):
    print([m[x[i][j]].as_long() for j in range (N)])
else:
  print("Não tem solução.")
```

```
== SOLUÇÃO ==
[4, 6, 1, 5, 3, 2]
[1, 5, 4, 2, 6, 3]
[2, 1, 3, 6, 5, 4]
[3, 2, 6, 1, 4, 5]
[5, 4, 2, 3, 1, 6]
[6, 3, 5, 4, 2, 1]
```

✓ 0 s concluído à(s) 16:32