A89474 Luís Filipe Cruz Sobral MIEI

```
import re
import numpy as np

def printTabuleiro(tabuleiro,sinaisL,size):
 l = 0
 while l < size:
  c = 0
  while c < size:
    print(tabuleiro[l][c],end = "")
    if c != (size - 1) and sinaisL[l][c] != 0:
     if(sinaisL[l][c] == 1):
      print(' > ',end = "")
     else:
      print(' < ',end = "")
    else:
     print('   ',end = "")
    c += 1

  print('')
  c = 0
  while c < size:
    if l != (size - 1) and sinaisC[l][c] != 0:
     if(sinaisC[l][c] == 1):
      print('^',end = "")
     else:
      print('V',end = "")
    else:
     print(' ',end = "")
    print(' ',end = "")
    c += 1

  print('')
  l+=1
 return

size = 0
tabRE = re.compile(r'P(\d+) (\d+) (\d+)')
sinaisRE = re.compile(r'S(-?\d+) (\d+) (\d+)')
sinaisCRE = re.compile(r'S2(-?\d+) (\d+) (\d+)')

with open("P1.txt") as f:
 file = f.read()
 s = re.search(r'Size (\d+)',file)
 size = int(s.group(1))

 linha = []
 for i in range(size):
   linha.append(0)

 tab = []
 for i in range(size):
```

```
      tab.append(linha.copy())

    sinaisL = np.zeros((size,size - 1), dtype=int)
    sinaisC = np.zeros((size - 1,size), dtype=int)

    tabContent = re.findall(tabRE, file)
    sinaisContent = re.findall(sinaisRE, file)
    sinaisCContent = re.findall(sinaisCRE, file)

    for line in tabContent:
     tab[int(line[0])][int(line[1])] = int(line[2])

    for line in sinaisContent:
     sinaisL[int(line[1])][int(line[2])] = int(line[0])

    for line in sinaisCContent:
     sinaisC[int(line[1])][int(line[2])] = int(line[0])

  printTabuleiro(tab,sinaisL,size)

  !pip install z3-solver
  from z3 import *

  X = [ [ Int("x_%s_%s" % (l, c)) for c in range(size) ]
        for l in range(size) ]

  cells_c  = [ And(1 <= X[l][c], X[l][c] <= size)
               for l in range(size) for c in range(size) ]

  rows_c   = [ Distinct(X[l]) for l in range(size) ]

  cols_c   = [ Distinct([ X[l][c] for l in range(size) ])
               for c in range(size) ]

  futoshiki = cells_c + rows_c + cols_c

  s = Solver()

  s.add(futoshiki)
  for l in range(size):
      for c in range(size - 1):
          if(sinaisL[l][c] == 1):
              s.add(X[l][c] > X[l][c + 1])
          elif (sinaisL[l][c] == -1):
              s.add(X[l][c] < X[l][c + 1])

  for l in range(size - 1):
      for c in range(size):
          if(sinaisC[l][c] == 1):
              s.add(X[l ][c] < X[l + 1][c])
          elif (sinaisC[l][c] == -1):
              s.add(X[l][c] > X[l + 1][c])

  for l in range(size):
      for c in range(size):
```

```
        if(tab[l][c] != 0):
          s.add(X[l][c] == tab[l][c])

if s.check() == sat:
    m = s.model()
    #print(m)
    tab = [ [ m.evaluate(X[l][c]) for c in range(size) ]
          for l in range(size) ]

    printTabuleiro(tab,sinaisL,size)
else:
    print("failed to solve")
```

```
0 > 0   0 > 0 > 0

4   0   0   0   2

0   0   4   0   0

0   0   0   0 < 4

0 < 0 < 0   0   0

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
5 > 4   3 > 2 > 1

4   3   1   5   2

2   1   4   3   5

3   5   2   1 < 4

1 < 2 < 5   4   3
```

✓ 7 s concluído à(s) 20:29