

▼ Grupo 24

Pedro Faria - A72640

Hugo Costeira - A87976

```
!pip install ortools
```

▼ Exercício 1

1. Pretende-se construir um horário semanal para o plano de reuniões de projeto de uma "StartUp" de acordo com as seguintes condições:
 1. Cada reunião ocupa uma sala (enumeradas 1...S) durante um "slot" (tempo,dia). Assume-se os dias enumerados 1..D e, em cada dia, os tempos enumerados 1..T.
 2. Cada reunião tem associado um projeto (enumerados 1..P) e um conjunto de participantes. Os diferentes colaboradores são enumerados 1..C.
 3. Cada projeto tem associado um conjunto de colaboradores, dos quais um é o líder. Cada projeto realiza um dado número de reuniões semanais. São "inputs" do problema o conjunto de colaboradores de cada projeto, o seu líder e o número de reuniões semanais.
 4. O líder do projeto participa em todas as reuniões do seu projeto; os restantes colaboradores podem ou não participar consoante a sua disponibilidade, num mínimo ("quorum") de 50 do total de colaboradores do projeto. A disponibilidade de cada participante, incluindo o líder, é um conjunto de "slots" ("inputs" do problema).

Começamos por importar a biblioteca de programação linear do OR-Tools e criar uma instância do solver e fazer a declaração da matriz de alocação X como um dicionário

```
from ortools.linear_solver import pywraplp
horario = pywraplp.Solver.CreateSolver('SCIP')
```

```
S,D,T,P,C=4,5,4,5,20
L,R=1,5
```

```
x = {}
for s in range(S):
    x[s] = {}
```

✓ 0s completed at 7:33 PM



```
x[s][d][t][p][c] = horario.BoolVar('x[%i][%i][%i][%i][%i]' % (s,d,t,p,c))
```

```
def X(s,d,t,p,c):                                # abreviatura
    return x[s][d][t][p][c]
```

1. Cada reunião ocupa uma sala (enumeradas 1..S) durante um “slot” (tempo,dia). Assume-se os dias enumerados 1..D e, em cada dia, os tempos enumerados 1..T.

Cada Projeto tem uma sala durante um slot:

$$\forall s < S \cdot \forall d < D \cdot \forall t < T \quad \sum_{p < P} x_{s,d,t,p,c} = 1$$

```
for s in range(S):
    for d in range(D):
        for p in range(P):
            horario.Add(sum([X(s,d,t,p,c) for p in range(P)]) == 1, name='Um')
```

2. Cada reunião tem associado um projeto (enumerados 1..P) e um conjunto de participantes. Os diferentes colaboradores são enumerados 1..C.

$$\forall p < P \quad \sum_{c < C} x_{s,d,t,p,c} \geq 1$$

```
for p in range(P):
    horario.Add(sum([X(s,d,t,p,c) for c in range(C)]) >= 1, name='Dois')
```

3. Cada projeto tem associado um conjunto de colaboradores, dos quais um é o líder. Cada projeto realiza um dado número de reuniões semanais. São “inputs” do problema o conjunto de colaboradores de cada projeto, o seu líder e o número de reuniões semanais.

L=Conjunto dos Lideres de Projeto

$$\forall p < P \cdot \forall c < C \quad \sum x_{s,d,t,p,c} = 1$$

colaboradores podem ou não participar consoante a sua disponibilidade, num mínimo (“quorum”) de 50% do total de colaboradores do projeto. A disponibilidade de cada participante, incluindo o líder, é um conjunto de “slots” (“inputs” do problema).

$$\forall_{p < P} \sum_{l < L} x_{s,d,t,p,c} = 1 \quad \wedge \quad \forall_{p < P} \sum_{c < C} x_{s,d,t,p,c} \geq C/2$$

```
for p in range(P):
    horario.Add(sum([X(s,d,t,p,c) for l in range(L)])==1)
    horario.Add(sum([X(s,d,t,p,c) for c in range(C)])>=(C/2))
```

Exercicio 2

Da definição do jogo “Sudoku” generalizado para a dimensão N ; o problema tradicional corresponde ao caso $N = 3$. O objetivo do Sudoku é preencher uma grelha de $N^2 \times N^2$ com inteiros positivos no intervalo 1 até N^2 , satisfazendo as seguintes regras

Cada inteiro no intervalo 1 até N^2 ocorre só uma vez em cada coluna, linha e secção $N \times N$.

No início do jogo uma fração $0 \leq \alpha < 1$ das N^4 casas da grelha são preenchidas de forma consistente com a regra anterior.

1. Construir um programa para inicializar a grelha a partir dos parâmetros N e α
2. Construir soluções do problema para as combinações de parâmetros $N \in \{3, 4, 5, 6\}$ e $\alpha \in \{0.0, 0.2, 0.4, 0.6\}$. Que conclusões pode tirar da complexidade computacional destas soluções.

```
def S(c,l,q):
    return s[l][c][q]

tabuleiro = ([[5, 4, 9, 0, 0, 1, 7, 3, 8],
               [3, 6, 7, 0, 0, 8, 0, 0, 1],
               [2, 0, 0, 0, 7, 3, 0, 4, 0],
               [0, 0, 0, 9, 0, 0, 0, 0, 5],
               [0, 0, 0, 7, 0, 5, 4, 6, 0],
               [1, 3, 5, 8, 4, 0, 0, 7, 0],
               [0, 0, 4, 0, 0, 0, 3, 0, 7],
               [7, 8, 0, 3, 5, 0, 0, 0, 6],
               [0, 2, 3, 0, 8, 0, 0, 0, 0]])
```

Cada inteiro no intervalo 1 até N² ocorre só uma vez em cada coluna, linha e secção N×N.

$$\forall_{1 \leq x \leq N^2} \sum_{c \leq N, l \leq N, q \leq N} s_{c,l,q} = 1$$

```
for x in range(1,N^2):
    sudoku.Add(sum([S(l,c,q) for l in range(N) for c in range(N) for q in range(N)]))
```

```
def criaTabuleiro(N,alpha):
    linha={}
    coluna{}
```

