Ex1_Horario

October 17, 2022

1 Exercício 1 (Horário) - Trabalho Prático 1

Grupo 4: Carlos Costa-A94543 Ruben Silva-A94633

2 Problema:

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" **1**..**T** (hora,dia).
    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.
    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.

São "inputs" do problema: i. Os parâmetros **S,T,P,C** ii. O conjunto de colaboradores de cada projeto, o seu líder e o número de reuniões semanais. iii. A disponibilidade de cada participante, incluindo o lider. Essa disponibilidade é um conjunto de "slots" representada numa matriz booleana de acessibilidade com uma linha por cada participante **1**..**C** e uma coluna por "slot" **1**..**T**. São critérios de optimização: i. Maximizar o número de reuniões efetivamente realizadas ii. Minimizar o número médio de reuniões por participante.

3 Análise do Problema

Este é um problema de alocação que requer uma matriz boleana para se resolver. Como tal, é necessário a utilização de variaveis binárias (1 ou 0). Para resolver este problema será necessárias S × T × D × P × C variáveis. Estas variáveis pertecerão à seguinte familia: $x_{s,t,d,p,c}$. $x_{s,t,d,p,c} = 1 \iff$ Colaborador estará presente na reunião p, sala s e no slot (t,d) Algumas variáveis que useramos para descrever as limitações e as obrigações do nosso problema: * $s \in \mathbf{S} = \{1..S\}; \rightarrow$ (Salas) * $t \in \mathbf{T} = \{1..T\}; \rightarrow$ (Slots de tempo = Horas) * $d \in \mathbf{D} = \{1..D\}; \rightarrow$ (Dias) * $p \in \mathbf{P} = \{1..P\}; \rightarrow$ (Projetos) * $c \in \mathbf{C} = \{1..C\}; \rightarrow$ (Colaboradores) * $y_p \in \mathbf{C} \rightarrow$ (Lider do projeto) * $n_p \rightarrow$ (Numero de reuniões semanais) * $X_p \subseteq \mathbf{C} \rightarrow$ (Grupo dos colaboradores do projeto) * $S_p \rightarrow$ (Lista de Slots de tempo disponiveis para um colaborador específico)

4 Limitações e obrigações

1. As salas só podem ter uma reunião a decorrer de cada vez É necessário que o somatório de todos os projetos para uma específica sala e num específico horário, seja no máximo 1, assim garantimos que cada sala só pode ter 1 uma reunião a decorrer.

$$\forall_{s\in\mathbf{S}}\forall_{t\in\mathbf{T}}\forall_{d\in\mathbf{D}} : \sum_{p=1}^{\mathbf{P}} x_{s,t,d,p,y_p} \leq 1$$

2. O lider tem de ter participado em todas as reuniões É necessário que o somatório de todos os horários e salas de um projeto p seja igual à sua pre-definida quantidade de reuniões (n_p).

$$\forall_{p\in\mathbf{P}} : \sum_{s=1}^{\mathbf{S}}\sum_{t=1}^{\mathbf{T}}\sum_{d=1}^{\mathbf{D}} x_{s,t,d,p,y_p} = n_p$$

3. Colaboradores só podem participar se tiverem disponibilidade É necessário que numa sala, horário e projeto específico, se o colaborador em questão não tenha esse horário disponível, então não poderá participar.

$$\forall_{s\in\mathbf{S}}\forall_{t\in\mathbf{T}}\forall_{d\in\mathbf{D}}\forall_{p\in\mathbf{P}}\forall_{c\in\mathbf{C}} : (t,d) \notin S_p \implies x_{s,t,d,p,c} = 0$$

4. Os Colaboradores só podem participar numa reuniao de cada vez É necessário que num horário, colaborador e projeto específico, o somátorio de todas as salas seja no máximo 1, assim garantimos que esse específico colaborador não estará noutra sala.

$$\forall_{t\in\mathbf{T}}\forall_{d\in\mathbf{D}}\forall_{p\in\mathbf{P}}\forall_{c\in\mathbf{C}} : \sum_{s=1}^{\mathbf{S}} x_{s,t,d,p,c} \leq 1$$

5. Colaboradores não podem participar em projetos que não sejam dele É necessário que numa sala, horário e projeto específico, se o colaborador em questão não estiver na lista dos colaboradores desse projeto específico, então nao pode participar.

$$\forall_{s\in\mathbf{S}}\forall_{t\in\mathbf{T}}\forall_{d\in\mathbf{D}}\forall_{p\in\mathbf{P}}\forall_{c\in\mathbf{C}} : c \notin X_p \implies x_{s,t,d,p,c} = 0$$

6. No minimo, é necessário 50% dos colaboradores nas reunioes ("quorum") É necessário que num horário e projeto específico, o somátorio de todas os colaboradores desse projeto tem de ser superior a 50% dos colaboradores alocados. **NOTA:** Foi usado a expressão ($x_{s,t,d,p,y_p} \times \frac{|X_p|}{2}$) porque assim, com a variável (x_{s,t,d,p,y_p}) conseguimos controlar a existencia ou não dessa reunião (0 se não existir, 1 se existir). Com a ausência dessa variável, estariamos a partir do pressuposto que a reunião existe sempre.

$$\forall_{s\in\mathbf{S}}\forall_{t\in\mathbf{T}}\forall_{d\in\mathbf{D}}\forall_{p\in\mathbf{P}} : \sum_{c=1}^{X_p} x_{s,t,d,p,c} \geq x_{s,t,d,p,y_p} \times \frac{|X_p|}{2}$$

7. O lider é obrigatório na reunião É necessário que num horário e projeto específico, o somátorio de todas os colaboradores desse projeto tem de ser no máximo igual ao somatório dos colaboradores. **NOTA:** Foi usado a expressão ($x_{s,t,d,p,y_p} \times |X_p|$) porque assim, com a variável (x_{s,t,d,p,y_p}) conseguimos controlar a existência do líder nela (0 se não existir, 1 se existir).

$$\forall_{s\in\mathbf{S}}\forall_{t\in\mathbf{T}}\forall_{d\in\mathbf{D}}\forall_{p\in\mathbf{P}} : \sum_{c=1}^{X_p} x_{s,t,d,p,c} \leq x_{s,t,d,p,y_p} \times |X_p|$$

5 Implementaçao do Problema

Importar o solver 1. Instalar o o ortools a partir da libraria do PyPi (pip) 2. Importar o pywraplp do ortools (contem o solver que iremos usar neste problema)

[ ]:
```
!pip install ortools
from ortools.linear_solver import pywraplp
```

```
Requirement already satisfied: ortools in
c:\users\ruben\appdata\local\programs\python\python310\lib\site-packages
(9.4.1874)
Requirement already satisfied: protobuf>=3.19.4 in
c:\users\ruben\appdata\local\programs\python\python310\lib\site-packages (from
ortools) (4.21.6)
Requirement already satisfied: absl-py>=0.13 in
c:\users\ruben\appdata\local\programs\python\python310\lib\site-packages (from
ortools) (1.2.0)
Requirement already satisfied: numpy>=1.13.3 in
c:\users\ruben\appdata\local\programs\python\python310\lib\site-packages (from
ortools) (1.23.3)
```

Resolver o codigo 1. É definido a função "horario" com return de: 0 ou um dicionário com todas as reuniões 1. Importar o Solver 2. Criar uma Lista de tuplos com o tipo do solver.BoolVar 3. Adicionar ao solver todas Limitações e Obrigações 4. Executar o solver

[ ]:
```
def horario(S, T, P, C, dados, disponibilidadeColabs):

    # Importar o solver
    solver = pywraplp.Solver.CreateSolver("SCIP")

    # Criar uma Lista de tuplos com o tipo do solver.BoolVar
    listVariaveis = {}
    D = 5
    for s in range(1, S+1):
        for t in range(1, T+1):
            for d in range(1, D+1):
                for p in range(1, P+1):
                    for c in range(1, C+1):
                        listVariaveis[s, t, d, p, c] = solver.BoolVar(
                            "s[%i]t[%i]d[%i]p[%i]c[%i]" % (s, t, d, p, c))

    # Limitação/Obrigação 1
    # As salas só podem ter uma reunião a decorrer de cada vez
    for s in range(1, S+1):
        for t in range(1, T+1):
            for d in range(1, D+1):
                solver.Add(
                    sum(listVariaveis[s, t, d, p, dados[p]["lider"]] for p in␣
↪range(1, P+1)) <= 1)
```

```
    # Limitação/Obrigação 2
    # O líder é obrigado a estar presente nas reunioes todas
    for p in range(1, P+1):
        solver.Add(sum(listVariaveis[s, t, d, p, dados[p]["lider"]] for t in␣
↪range(
            1, T+1) for d in range(1, D+1) for s in range(1, S+1)) ==␣
↪dados[p]["nReunioesSemanais"])

    # Limitação/Obrigação 3
    # Colaboradores só podem participar se tiverem disponibilidade
    for s in range(1, S+1):
        for t in range(1, T+1):
            for d in range(1, D+1):
                for p in range(1, P+1):
                    for c in range(1, C+1):
                        if (t, d) not in disponibilidadeColabs[c]:
                            solver.Add(listVariaveis[s, t, d, p, c] == 0)

    # Limitação/Obrigação 4
    # Os Colaboradores só podem participar numa reuniao de cada vez
    for t in range(1, T+1):
        for d in range(1, D+1):
            for p in range(1, P+1):
                for c in range(1, C+1):
                    solver.Add(sum(listVariaveis[s, t, d, p, c]
                                   for s in range(1, S+1)) <= 1)

    # Limitação/Obrigação 5
    # Colaboradores não podem participar em projetos que não sejam dele
    for s in range(1, S+1):
        for t in range(1, T+1):
            for d in range(1, D+1):
                for p in range(1, P+1):
                    for c in range(1, C+1):
                        if c not in dados[p]["colaboradores"]:
                            solver.Add(listVariaveis[s, t, d, p, c] == 0)

    # Limitação/Obrigação 6
    # No minimo, é necessário 50% dos colaboradores nas reunioes ("quorum")
    for s in range(1, S+1):
        for t in range(1, T+1):
            for d in range(1, D+1):
                for p in range(1, P+1):
                    solver.Add(sum(listVariaveis[s, t, d, p, c] for c in␣
↪dados[p]["colaboradores"]) >=
```

```
                                (listVariaveis[s, t, d, p,␣
↪dados[p]["lider"]]*(len(dados[p]["colaboradores"])/2)))

    # Limitação/Obrigação 7
    # O lider é obrigatorio na reuniao
    for s in range(1, S+1):
        for t in range(1, T+1):
            for d in range(1, D+1):
                for p in range(1, P+1):
                    solver.Add((sum(listVariaveis[s, t, d, p, c] for c in␣
↪dados[p]["colaboradores"]))
                                <= listVariaveis[s, t, d, p, dados[p]["lider"]]␣
↪* len(dados[p]["colaboradores"]))

    # Executar o solver
    status = solver.Solve()
    if status == solver.OPTIMAL:
        final = []
        for s in range(1, S+1):
            for t in range(1, T+1):
                for d in range(1, D+1):
                    for p in range(1, P+1):
                        auxListColaboradores = []

                        # Retornar os dados finais para usar posteriormente␣
↪numa tabela
                        for c in range(1, C+1):
                            if listVariaveis[s, t, d, p, c].solution_value():
                                auxListColaboradores.append(c)
                        if auxListColaboradores:
                            #final = [((projeto,dia),Dicionario com dados).....]
                            #O projeto e o dia foram colocados no inicio para␣
↪um sort na função seguinte
                            final.append(((p,d),{
                                "projeto": p,
                                "horario": (t, d),
                                "lider": dados[p]["lider"],
                                "sala": s,
                                "colaboradores": sorted(auxListColaboradores)
                            }))
        return final
    else:
        return 0
```

Funções para gerar parametros

[ ]:
```
# 2 Dicionarios e 1 Set pra guardar toda a informação essencial
disponibilidadeColabs = {}
dadosProjetos = {}
slots = set()

# Esta função permite guardar todos os dados de cada projeto
def adicionar_dados_projetos(nr, colabs, lider, nrReunioesSemanais):
    dadosProjetos[nr] = {
        "colaboradores": colabs,
        "lider": lider,
        "nReunioesSemanais": nrReunioesSemanais}

# Esta funcao cria todas as slots disponiveis
def todas_slots(T):
    for t in range(1, T+1):
        for d in range(1, 6):
            if (t, d) not in slots:
                slots.add((t, d))
```

Exemplo de um horário 1 (Impossivel) (usamos este exemplo debaixo para testar alguns absurdos e certificar que não funcionam)

[ ]:
```
# Parametros iniciais:
S = 3
T = 8  # usaremos o T como se fosse horas num dia
P = 4
C = 7

# Gerar todas as slots de Tempo de dias
todas_slots(T)

# Adicionar os dados dos projetos
adicionar_dados_projetos(1, {1, 6, 7}, 1, 6)
adicionar_dados_projetos(2, {1, 4}, 4, 4)
adicionar_dados_projetos(3, { 5, 6, 7}, 1, 8) #absurdo pois o lider nao é␣
 ↪colaborador
adicionar_dados_projetos(4, {1,2, 3, 4, 5, 6, 7}, 1, 5)


disponibilidadeColabs[1] = {(t, d)for (t, d) in slots if t <= 2}
disponibilidadeColabs[2] = {(t, d)for (t, d) in slots if t <= 3}
disponibilidadeColabs[3] = {(t, d)for (t, d) in slots if t >= 7}
disponibilidadeColabs[4] = {(t, d)for (t, d) in slots if t <= 5 and d != 5}
disponibilidadeColabs[5] = {(t, d)for (t, d) in slots if d ==3}
disponibilidadeColabs[6] = {(t, d)for (t, d) in slots if d == 6 and t !=5}␣
 ↪#absurdo pois nao existe d==6
disponibilidadeColabs[7] = {(t, d)for (t, d) in slots if d<=3}
```

Exemplo de um horário 2 (Possível) (Exemplo bastante simples)

[ ]:
```
# Parametros iniciais:
S = 3
T = 8
P = 2
C = 4

todas_slots(T)

adicionar_dados_projetos(1, {1, 2,3}, 2, 6)
adicionar_dados_projetos(2, {3, 2}, 3, 5)


disponibilidadeColabs[1] = {(t, d)for (t, d) in slots if t <= 2}
disponibilidadeColabs[2] = {(t, d)for (t, d) in slots if t <= 3}
disponibilidadeColabs[3] = {(t, d)for (t, d) in slots if t >= 7}
disponibilidadeColabs[4] = {(t, d)for (t, d) in slots if t <= 5 and d != 5}
```

Exemplo de um horário 2 (Possível) (Um exemplo bem mais complexo)

[ ]:
```
# Parametros iniciais:
S = 5
T = 10
P = 8
C = 15

todas_slots(T)

adicionar_dados_projetos(1, {1, 2,3,5,6,7,12,15}, 2, 6)
adicionar_dados_projetos(2, {3, 2,8,9,13,14,15}, 8, 5)
adicionar_dados_projetos(3, {1, 2,13,6,5,7}, 2, 15)
adicionar_dados_projetos(4, {3, 2,8,5,}, 3, 5)
adicionar_dados_projetos(5, {1, 2,3,11,8,7,4}, 1, 4)
adicionar_dados_projetos(6, {3, 2}, 3, 5)
adicionar_dados_projetos(7, {1,2}, 2, 9)
adicionar_dados_projetos(8, {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}, 3, 5)


disponibilidadeColabs[1] = {(t, d)for (t, d) in slots if t <= 2}
disponibilidadeColabs[2] = {(t, d)for (t, d) in slots if t <= 3}
disponibilidadeColabs[3] = {(t, d)for (t, d) in slots if t >= 4}
disponibilidadeColabs[4] = {(t, d)for (t, d) in slots if t <= 5 and d != 5}
disponibilidadeColabs[5] = {(t, d)for (t, d) in slots if t <= 2}
disponibilidadeColabs[6] = {(t, d)for (t, d) in slots if t <= 3}
disponibilidadeColabs[7] = {(t, d)for (t, d) in slots if t >= 3}
disponibilidadeColabs[8] = {(t, d)for (t, d) in slots}
disponibilidadeColabs[9] = {(t, d)for (t, d) in slots }
```

```
disponibilidadeColabs[10] = {(t, d)for (t, d) in slots if t <= 3}
disponibilidadeColabs[11] = {(t, d)for (t, d) in slots}
disponibilidadeColabs[12] = {(t, d)for (t, d) in slots if t <= 5 and d != 5}
disponibilidadeColabs[13] = {(t, d)for (t, d) in slots if d <= 2}
disponibilidadeColabs[14] = {(t, d)for (t, d) in slots if d !=1}
disponibilidadeColabs[15] = {(t, d)for (t, d) in slots if t >= 7}
```

Tabela Final

[ ]:
```
def tabela_final():
    dadosHorario = horario(S, T, P, C, dadosProjetos, disponibilidadeColabs)
    if dadosHorario== 0:
        print("Horario Impossivel")
    else:
        dadosHorario=sorted(dadosHorario, key=lambda i: (i[0][0],i[0][1]))
        for ((proj,dia),dictHor) in dadosHorario:
            print("Projeto: "+str(proj)+"\nSlot (hora,dia): ("+str(dia)+" ,␣
 ↪"+str(dictHor["horario"][0])+")"+"\nLider: "+str(dictHor["lider"]))
            print("Colaboradores: "+str(dictHor["colaboradores"])+"\nSala:␣
 ↪"+str(dictHor["sala"]))
            ␣
 ↪print("_____")
```

Executar o Código

[ ]:
```
tabela_final()
```

```
Projeto: 1
Slot (hora,dia): (1 , 2)
Lider: 2
Colaboradores: [1, 2, 5, 6, 12]
Sala: 1

--------------------------------------------------------------------------------
Projeto: 1
Slot (hora,dia): (1 , 3)
Lider: 2
Colaboradores: [2, 6, 7, 12]
Sala: 1

--------------------------------------------------------------------------------
Projeto: 1
Slot (hora,dia): (3 , 1)
Lider: 2
Colaboradores: [1, 2, 5, 6, 12]
Sala: 1

--------------------------------------------------------------------------------
Projeto: 1
Slot (hora,dia): (3 , 3)
Lider: 2
```

```
Colaboradores: [2, 6, 7, 12]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 1
Slot (hora,dia): (5 , 1)
Lider: 2
Colaboradores: [1, 2, 5, 6]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 1
Slot (hora,dia): (5 , 2)
Lider: 2
Colaboradores: [1, 2, 5, 6]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 2
Slot (hora,dia): (1 , 8)
Lider: 8
Colaboradores: [3, 8, 9, 13, 15]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 2
Slot (hora,dia): (1 , 10)
Lider: 8
Colaboradores: [3, 8, 9, 13, 15]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 2
Slot (hora,dia): (2 , 5)
Lider: 8
Colaboradores: [3, 8, 9, 13, 14]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 2
Slot (hora,dia): (3 , 7)
Lider: 8
Colaboradores: [3, 8, 9, 14, 15]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 2
Slot (hora,dia): (4 , 10)
Lider: 8
Colaboradores: [3, 8, 9, 14, 15]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (1 , 1)
Lider: 2
```

```
Colaboradores: [1, 2, 5, 6, 13]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (1 , 2)
Lider: 2
Colaboradores: [1, 2, 5, 6, 13]
Sala: 3
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (1 , 3)
Lider: 2
Colaboradores: [2, 6, 7, 13]
Sala: 3
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (2 , 1)
Lider: 2
Colaboradores: [1, 2, 5, 6, 13]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (2 , 2)
Lider: 2
Colaboradores: [1, 2, 5, 6, 13]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (2 , 3)
Lider: 2
Colaboradores: [2, 6, 7, 13]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (3 , 1)
Lider: 2
Colaboradores: [1, 2, 5, 6]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (3 , 2)
Lider: 2
Colaboradores: [1, 2, 5, 6]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (3 , 3)
Lider: 2
```

```
Colaboradores: [2, 6, 7]
Sala: 3
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (4 , 1)
Lider: 2
Colaboradores: [1, 2, 5, 6]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (4 , 2)
Lider: 2
Colaboradores: [1, 2, 5, 6]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (4 , 3)
Lider: 2
Colaboradores: [2, 6, 7]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (5 , 1)
Lider: 2
Colaboradores: [1, 2, 5, 6]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (5 , 2)
Lider: 2
Colaboradores: [1, 2, 5, 6]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 3
Slot (hora,dia): (5 , 3)
Lider: 2
Colaboradores: [2, 6, 7]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 4
Slot (hora,dia): (2 , 9)
Lider: 3
Colaboradores: [3, 8]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 4
Slot (hora,dia): (4 , 5)
Lider: 3
```

```
Colaboradores: [3, 8]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 4
Slot (hora,dia): (4 , 6)
Lider: 3
Colaboradores: [3, 8]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 4
Slot (hora,dia): (5 , 7)
Lider: 3
Colaboradores: [3, 8]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 4
Slot (hora,dia): (5 , 8)
Lider: 3
Colaboradores: [3, 8]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 5
Slot (hora,dia): (2 , 1)
Lider: 1
Colaboradores: [1, 2, 4, 8, 11]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 5
Slot (hora,dia): (2 , 2)
Lider: 1
Colaboradores: [1, 2, 4, 8, 11]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 5
Slot (hora,dia): (3 , 2)
Lider: 1
Colaboradores: [1, 2, 4, 8, 11]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 5
Slot (hora,dia): (4 , 1)
Lider: 1
Colaboradores: [1, 2, 4, 8, 11]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 6
Slot (hora,dia): (2 , 8)
Lider: 3
```

```
Colaboradores: [3]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 6
Slot (hora,dia): (3 , 4)
Lider: 3
Colaboradores: [3]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 6
Slot (hora,dia): (3 , 8)
Lider: 3
Colaboradores: [3]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 6
Slot (hora,dia): (4 , 9)
Lider: 3
Colaboradores: [3]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 6
Slot (hora,dia): (5 , 9)
Lider: 3
Colaboradores: [3]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (1 , 1)
Lider: 2
Colaboradores: [2]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (1 , 2)
Lider: 2
Colaboradores: [2]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (1 , 3)
Lider: 2
Colaboradores: [2]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (2 , 3)
Lider: 2
```

```
Colaboradores: [2]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (3 , 3)
Lider: 2
Colaboradores: [2]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (3 , 1)
Lider: 2
Colaboradores: [2]
Sala: 3
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (4 , 2)
Lider: 2
Colaboradores: [2]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (4 , 3)
Lider: 2
Colaboradores: [2]
Sala: 2
--------------------------------------------------------------------------------
Projeto: 7
Slot (hora,dia): (5 , 3)
Lider: 2
Colaboradores: [2]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 8
Slot (hora,dia): (1 , 5)
Lider: 3
Colaboradores: [3, 4, 7, 8, 9, 11, 12, 13]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 8
Slot (hora,dia): (2 , 4)
Lider: 3
Colaboradores: [3, 4, 7, 8, 9, 11, 12, 13, 14]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 8
Slot (hora,dia): (2 , 10)
Lider: 3
```

```
Colaboradores: [3, 7, 8, 9, 11, 13, 14, 15]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 8
Slot (hora,dia): (3 , 5)
Lider: 3
Colaboradores: [3, 4, 7, 8, 9, 11, 12, 14]
Sala: 1
--------------------------------------------------------------------------------
Projeto: 8
Slot (hora,dia): (4 , 4)
Lider: 3
Colaboradores: [3, 4, 7, 8, 9, 11, 12, 14]
Sala: 1
--------------------------------------------------------------------------------
```