In [1]:

```
import numpy as np
from gurobipy import *
```

In [2]:

```
T = 5
m = Model()
x = m.addVars(T, lb=np.zeros(5), vtype = GRB.BINARY, name = "if_chosen")
m.setObjective(quicksum(x[t] for t in range(T)), GRB.MINIMIZE)

c1 = m.addConstr(x[0] + x[1] >= 1)
c2 = m.addConstr(x[0] + x[3] >= 1)
c3 = m.addConstr(x[2] >= 1)
c4 = m.addConstr(x[4] >= 1)
c5 = m.addConstr(x[1] + x[4] >= 1)
c6 = m.addConstr(x[1] + x[2] >= 1)
c7 = m.addConstr(x[3] >= 1)
c8 = m.addConstr(x[0] + x[3] >= 1)
```

```
Restricted license - for non-production use only - expires 2022-01-13
```

In [3]:

```
m.optimize()
m.printAttr('X')
```

```
Gurobi Optimizer version 9.1.2 build v9.1.2rc0 (win64)
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads
Optimize a model with 8 rows, 5 columns and 13 nonzeros
Model fingerprint: 0x74c2bfc5
Variable types: 0 continuous, 5 integer (5 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 1e+00]
  RHS range        [1e+00, 1e+00]
Found heuristic solution: objective 4.0000000
Presolve removed 8 rows and 5 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds
Thread count was 1 (of 8 available processors)

Solution count 1: 4

Optimal solution found (tolerance 1.00e-04)
Best objective 4.000000000000e+00, best bound 4.000000000000e+00, gap 0.0000%

    Variable            X
-------------------------
if_chosen[1]            1
if_chosen[2]            1
if_chosen[3]            1
if_chosen[4]            1
```