📝

OR | HW2

Status	In progress
Due date	@April 15, 2023
Type	Assignments

```
from gurobipy import *
import pandas as pd
import numpy as np
import grblogtools as glt

df = pd.read_csv('OR_hw02_data.csv', header=None)
print(df.to_string())

p = 8 # build at most p parks
n, m = df.shape
print(df.shape) # (m=60, n=20)
towns  = range(m)
potential_locations = range(n)

eg4d = Model("eg4d")

#-------- Add variables as a list ---------#
# xj = 1 if a park is built at loc j
x = []
for j in potential_locations:
    x.append(eg4d.addVar(lb=0, vtype = GRB.BINARY, name = "x" + str(j+1)))

# yij = 1 if the park in location j is the closest one for town i
y = []
for i in towns:
    y.append([])
    for j in potential_locations:
        y[i].append(eg4d.addVar(lb = 0, vtype = GRB.BINARY, name = "y" + str(i+1) + str(j+1)))

# dij = the distance between town i and location j
d = []
for i in towns:
    d.append([])
    for j in potential_locations:
        d[i].append(eg4d.addVar(lb = df.iloc[j, i], vtype = GRB.INTEGER, name = "d" + str(i+1) + str(j+1)))

# build at most p parks in potential locations.
```

```
p = eg4d.addVar(lb = 0, vtype = GRB.INTEGER, name = "p")

# w = the maximum distance for each people to move to her/his closest park.
w = eg4d.addVar(lb = 0, vtype = GRB.INTEGER, name = "max_distance")

eg4d.setObjective(w,GRB.MINIMIZE)

eg4d.addConstrs((quicksum(y[i][j] for j in potential_locations) == 1 for i in towns),
 "每個town都有一個最近的park")
eg4d.addConstrs((y[i][j] <= x[j] for i in towns for j in potential_locations), "要確定
那個park有蓋")
eg4d.addConstr((quicksum(x[j] for j in potential_locations) <= p), "最多蓋p個park")
eg4d.addConstrs((d[i][j] * y[i][j] <= w for i in towns for j in potential_locations),
 "每個town到最近的park的最遠距離")
#eg4d.addConstrs((quicksum(d[i][j] * y[i][j] for j in potential_locations) <= w for i
 in towns), "每個town到最近的park的最遠距離")

eg4d.optimize()
print("z* = ", eg4d.ObjVal)
```

```
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)

CPU model: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, instruction set [SSE2|AVX|AVX2|A
VX512]
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 1261 rows, 2422 columns and 3621 nonzeros
Model fingerprint: 0xfb78fc48
Model has 1200 quadratic constraints
Variable types: 0 continuous, 2422 integer (1220 binary)
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  QMatrix range    [1e+00, 1e+00]
  QLMatrix range   [1e+00, 1e+00]
  Objective range  [1e+00, 1e+00]
  Bounds range     [1e+00, 5e+02]
  RHS range        [1e+00, 1e+00]
Presolve removed 1201 rows and 21 columns
Presolve time: 0.01s
Presolved: 3660 rows, 6001 columns, 9600 nonzeros
Presolved model has 2400 SOS constraint(s)
Variable types: 0 continuous, 6001 integer (2400 binary)
Found heuristic solution: objective 456.0000000
Found heuristic solution: objective 447.0000000
Found heuristic solution: objective 340.0000000
Found heuristic solution: objective 272.0000000

Root relaxation: objective 0.000000e+00, 2182 iterations, 0.01 seconds (0.00 work unit
s)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0    0.00000    0   60  272.00000    0.00000   100%     -    0s
H    0     0                      267.0000000    0.00000   100%     -    0s
```

```
H    0     0                     265.0000000    0.00000   100%     -    0s
H    0     0                     264.0000000    0.00000   100%     -    0s

Explored 1 nodes (2182 simplex iterations) in 0.34 seconds (0.09 work units)
Thread count was 8 (of 8 available processors)

Solution count 7: 264 265 267 ... 456

Optimal solution found (tolerance 1.00e-04)
Best objective 2.640000000000e+02, best bound 2.640000000000e+02, gap 0.0000%
z* =  264.0
```