In [2]:
```
## !pip install pulp
```

```
Collecting pulp
  Downloading PuLP-2.6.0-py3-none-any.whl (14.2 MB)
Installing collected packages: pulp
Successfully installed pulp-2.6.0
```

In [1]:
```
from pulp import LpMaximize, LpMinimize, LpProblem, LpStatus, lpSum, LpVariable
import pandas as pd
```

A enviromental firm is planning a collection project in order to add to their inventory of algae samples. They can order any number of algae bloom samples from any of seven lakes in the region. The cost per sample from each lake is shown in Table 1 below. Each lake also has its own balance of algae populations. The algae population composition for each lake is contained in the data file AlgaeData.csv For example, a sample from Lake Barkley has a 14% probability of being identified in the lab as Type C and 18% probability of Type D, etc. The firm wants to be sure that in the end they have at least 1000 algal units for each algae type. They also want to make sure that no more than 20% of the samples come from any one lake. How many samples should the firm order from each lake in order to minimize their total cost and meet their requirements?

Table 1:

Lake	Cost of Sample
Lake Barkley	5.00
Lake Alhoun	3.50
Norris Lake	6.50
Lake Powell	10.00
Hop Brook Reservoir	2.50
Highland Pond	5.50
Lake Saltonstall	7.00

In [22]:
```
data = pd.read_csv('MidtermReview_AlgaeData(1).csv')
data = data.set_index('Algae')
data
```

Out[22]:

Algae	Lake_Barkley	Lake_Alhoun	Norris_Lake	Lake_Powell	Hop_Brook_Reservoir	Highland_Po
Type_A	0.00	0.14	0.01	0.12	0.02	0.
Type_B	0.01	0.14	0.17	0.14	0.00	0.
Type_C	0.14	0.05	0.06	0.01	0.07	0.
Type_D	0.18	0.17	0.11	0.08	0.10	0.
Type_E	0.17	0.06	0.04	0.06	0.12	0.
Type_F	0.13	0.00	0.14	0.08	0.07	0.
Type_G	0.06	0.01	0.06	0.16	0.09	0.
Type_H	0.12	0.14	0.02	0.12	0.00	0.
Type_I	0.03	0.08	0.14	0.20	0.14	0.

In [41]:
```
algae = data.index
lakes = data.columns
print(algae)
print(lakes)
required = 1000

costs = dict(zip([l for l in lakes], [5, 3.5, 6.5, 10, 2.5, 5.5, 7]))
costs
```

```
Index(['Type_A', 'Type_B', 'Type_C', 'Type_D', 'Type_E', 'Type_F', 'Type_G',
       'Type_H', 'Type_I'],
      dtype='object', name='Algae')
Index(['Lake_Barkley', 'Lake_Alhoun', 'Norris_Lake', 'Lake_Powell',
       'Hop_Brook_Reservoir', 'Highland_Pond', 'Lake_Saltonstall'],
      dtype='object')
```

Out[41]:
```
{'Lake_Barkley': 5,
 'Lake_Alhoun': 3.5,
 'Norris_Lake': 6.5,
 'Lake_Powell': 10,
 'Hop_Brook_Reservoir': 2.5,
 'Highland_Pond': 5.5,
 'Lake_Saltonstall': 7}
```

In [35]:
```
model = LpProblem("Lake_Samples", LpMinimize)
```

In [42]:
```
# Construct Decision Variables

samples = LpVariable.dicts("sample", (l for l in lakes), lowBound = 0, cat = 'Int
samples
```

Out[42]:
```
{'Lake_Barkley': sample_Lake_Barkley,
 'Lake_Alhoun': sample_Lake_Alhoun,
 'Norris_Lake': sample_Norris_Lake,
 'Lake_Powell': sample_Lake_Powell,
 'Hop_Brook_Reservoir': sample_Hop_Brook_Reservoir,
 'Highland_Pond': sample_Highland_Pond,
 'Lake_Saltonstall': sample_Lake_Saltonstall}
```

In [43]:
```
# Add the objective function to the model
model += lpSum(costs[l] * samples[l] for l in lakes)
```

In [44]:
```
# Add the constraints to the model

for a in algae:
    model += lpSum(samples[l] * data.loc[a, l] for l in lakes) >= required

for l in lakes:
    model += samples[l] <= .2 * lpSum(samples[l] for l in lakes)
```

In [10]:
```
model.solve()
```

Out[10]:
```
1
```

In [11]:
```
model.solve()

LpStatus[model.status]
```

Out[11]:
```
'Optimal'
```

In [12]:
```
model.objective.value()
```

Out[12]:
```
79768.5
```

In [13]:
```
for v in model.variables(): print(f"{v.name}: {v.varValue}")
```

```
sample_Highland_Pond: 3053.0
sample_Hop_Brook_Reservoir: 3057.0
sample_Lake_Alhoun: 3057.0
sample_Lake_Barkley: 3057.0
sample_Lake_Powell: 2701.0
sample_Lake_Saltonstall: 0.0
sample_Norris_Lake: 360.0
```

In [9]:
```
model
```

Out[9]:
```
Lake_Samples:
MINIMIZE
5.5*sample_Highland_Pond + 2.5*sample_Hop_Brook_Reservoir + 3.5*sample_Lake_
Alhoun + 5*sample_Lake_Barkley + 10*sample_Lake_Powell + 7*sample_Lake_Salto
nstall + 6.5*sample_Norris_Lake + 0.0
SUBJECT TO
_C1: 0.06 sample_Highland_Pond + 0.02 sample_Hop_Brook_Reservoir
 + 0.14 sample_Lake_Alhoun + 0.12 sample_Lake_Powell
 + 0.1 sample_Lake_Saltonstall + 0.01 sample_Norris_Lake >= 1000

_C2: 0.19 sample_Highland_Pond + 0.14 sample_Lake_Alhoun
 + 0.01 sample_Lake_Barkley + 0.14 sample_Lake_Powell
 + 0.19 sample_Lake_Saltonstall + 0.17 sample_Norris_Lake >= 1000

_C3: 0.09 sample_Highland_Pond + 0.07 sample_Hop_Brook_Reservoir
 + 0.05 sample_Lake_Alhoun + 0.14 sample_Lake_Barkley
 + 0.01 sample_Lake_Powell + 0.07 sample_Lake_Saltonstall
 + 0.06 sample_Norris_Lake >= 1000

_C4: 0.07 sample_Highland_Pond + 0.1 sample_Hop_Brook_Reservoir
 + 0.17 sample_Lake_Alhoun + 0.18 sample_Lake_Barkley
 + 0.08 sample_Lake_Powell + 0.16 sample_Lake_Saltonstall
 + 0.11 sample_Norris_Lake >= 1000

_C5: 0.08 sample_Highland_Pond + 0.12 sample_Hop_Brook_Reservoir
 + 0.06 sample_Lake_Alhoun + 0.17 sample_Lake_Barkley
 + 0.06 sample_Lake_Powell + 0.18 sample_Lake_Saltonstall
 + 0.04 sample_Norris_Lake >= 1000

_C6: 0.04 sample_Highland_Pond + 0.07 sample_Hop_Brook_Reservoir
 + 0.13 sample_Lake_Barkley + 0.08 sample_Lake_Powell
 + 0.14 sample_Norris_Lake >= 1000

_C7: 0.18 sample_Highland_Pond + 0.09 sample_Hop_Brook_Reservoir
 + 0.01 sample_Lake_Alhoun + 0.06 sample_Lake_Barkley
 + 0.16 sample_Lake_Powell + 0.1 sample_Lake_Saltonstall
 + 0.06 sample_Norris_Lake >= 1000

_C8: 0.2 sample_Highland_Pond + 0.14 sample_Lake_Alhoun
 + 0.12 sample_Lake_Barkley + 0.12 sample_Lake_Powell
 + 0.02 sample_Lake_Saltonstall + 0.02 sample_Norris_Lake >= 1000

_C9: 0.02 sample_Highland_Pond + 0.14 sample_Hop_Brook_Reservoir
 + 0.08 sample_Lake_Alhoun + 0.03 sample_Lake_Barkley + 0.2 sample_Lake_Powe
ll
 + 0.16 sample_Lake_Saltonstall + 0.14 sample_Norris_Lake >= 1000

_C10: - 0.2 sample_Highland_Pond - 0.2 sample_Hop_Brook_Reservoir
 - 0.2 sample_Lake_Alhoun + 0.8 sample_Lake_Barkley - 0.2 sample_Lake_Powell
 - 0.2 sample_Lake_Saltonstall - 0.2 sample_Norris_Lake <= 0

_C11: - 0.2 sample_Highland_Pond - 0.2 sample_Hop_Brook_Reservoir
 + 0.8 sample_Lake_Alhoun - 0.2 sample_Lake_Barkley - 0.2 sample_Lake_Powell
 - 0.2 sample_Lake_Saltonstall - 0.2 sample_Norris_Lake <= 0

_C12: - 0.2 sample_Highland_Pond - 0.2 sample_Hop_Brook_Reservoir
 - 0.2 sample_Lake_Alhoun - 0.2 sample_Lake_Barkley - 0.2 sample_Lake_Powell
 - 0.2 sample_Lake_Saltonstall + 0.8 sample_Norris_Lake <= 0

_C13: - 0.2 sample_Highland_Pond - 0.2 sample_Hop_Brook_Reservoir
 - 0.2 sample_Lake_Alhoun - 0.2 sample_Lake_Barkley + 0.8 sample_Lake_Powell
 - 0.2 sample_Lake_Saltonstall - 0.2 sample_Norris_Lake <= 0

_C14: - 0.2 sample_Highland_Pond + 0.8 sample_Hop_Brook_Reservoir
 - 0.2 sample_Lake_Alhoun - 0.2 sample_Lake_Barkley - 0.2 sample_Lake_Powell
 - 0.2 sample_Lake_Saltonstall - 0.2 sample_Norris_Lake <= 0

_C15: 0.8 sample_Highland_Pond - 0.2 sample_Hop_Brook_Reservoir
 - 0.2 sample_Lake_Alhoun - 0.2 sample_Lake_Barkley - 0.2 sample_Lake_Powell
 - 0.2 sample_Lake_Saltonstall - 0.2 sample_Norris_Lake <= 0
```

```
_C16: - 0.2 sample_Highland_Pond - 0.2 sample_Hop_Brook_Reservoir
 - 0.2 sample_Lake_Alhoun - 0.2 sample_Lake_Barkley - 0.2 sample_Lake_Powell
 + 0.8 sample_Lake_Saltonstall - 0.2 sample_Norris_Lake <= 0

VARIABLES
0 <= sample_Highland_Pond Integer
0 <= sample_Hop_Brook_Reservoir Integer
0 <= sample_Lake_Alhoun Integer
0 <= sample_Lake_Barkley Integer
0 <= sample_Lake_Powell Integer
0 <= sample_Lake_Saltonstall Integer
0 <= sample_Norris_Lake Integer
```

In [ ]: