MP-2 Tutorial - 6

Problem 3

In [1]:
```
!pip install python-constraint
```

```
Requirement already satisfied: python-constraint in /srv/conda/envs/notebook/lib/python3.7/site-packages (1.4.0)
```

In [2]:
```
import constraint
```

In [3]:
```
problem = constraint.Problem()

problem.addVariable('A', range(31))
problem.addVariable('B', range(45))
problem.addVariable('C', range(76))
problem.addVariable('D', range(101))
```

In [4]:
```
def weight_constraint(a, b, c, d):
    if (a*100 + b*45 + c*10 + d*25) <= 3000:
        return True
```

In [5]:
```
def volume_constraint(a, b, c, d):
    if (a*8*2.5*0.5 + b*6*2*0.5 * c*2*2*0.5 + d*3*3*0.5) <= 1000:
        return True
```

In [6]:
```
def value_constraint(a, b, c, d):
    if (a*8 + b*6.8 + c*4 + d*3) < 300:
        return True
```

In [7]:
```
problem.addConstraint(weight_constraint, "ABCD")
problem.addConstraint(volume_constraint, "ABCD")
problem.addConstraint(value_constraint, "ABCD")
```

In [8]:
```
maximum_sweetness = 0
solution_found = {}
solutions = problem.getSolutions()
```

In [9]:
```
for s in solutions:
    current_sweetness = s['A']*10 + s['B']*8 + s['C']*4.5 + s['D']*3.5
    if current_sweetness > maximum_sweetness:
        maximum_sweetness = current_sweetness
        solution_found = s
print("""
The maximum sweetness we can bring is: {}
We'll bring:
{} A Chocolates,
{} B Chocolates,
{} C Chocolates,
{} D Chocolates
""".format(maximum_sweetness, solution_found['A'], solution_found['B'], solution_found['C'], solution_found['D']))
```

```
The maximum sweetness we can bring is: 365.0
We'll bring:
27 A Chocolates,
2 B Chocolates,
16 C Chocolates,
2 D Chocolates
```