MP-2 Tutorial - 6

190031920

Avuthu Nikhil Reddy

Problem 1:

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

```
Collecting python-constraint
  Downloading python-constraint-1.4.0.tar.bz2 (18 kB)
Building wheels for collected packages: python-constraint
  Building wheel for python-constraint (setup.py) ... done
  Created wheel for python-constraint: filename=python_constraint-1.4.0-py2.py3-none-any.whl size=2
4080 sha256=c1f872c6ebf587d6e1c078acf3a17d317e194818d4473abe3842d493f37b228b
  Stored in directory: /home/jovyan/.cache/pip/wheels/07/27/db/1222c80eb1e431f3d2199c12569cb1cac60f
562a451fe30479
Successfully built python-constraint
Installing collected packages: python-constraint
Successfully installed python-constraint-1.4.0
```

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

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

problem.addVariable('x', [1,2,3])
problem.addVariable('y', range(10))
```

In [4]:
```
def our_constraint(x, y):
    if x + y >= 5:
        return True
```

In [5]:
```
problem.addConstraint(our_constraint, ['x','y'])

solutions = problem.getSolutions()
```

In [6]:
```
for solution in solutions:
    print(solution)
```

```
{'x': 3, 'y': 9}
{'x': 3, 'y': 8}
{'x': 3, 'y': 7}
{'x': 3, 'y': 6}
{'x': 3, 'y': 5}
{'x': 3, 'y': 4}
{'x': 3, 'y': 3}
{'x': 3, 'y': 2}
{'x': 2, 'y': 9}
{'x': 2, 'y': 8}
{'x': 2, 'y': 7}
{'x': 2, 'y': 6}
{'x': 2, 'y': 5}
{'x': 2, 'y': 4}
{'x': 2, 'y': 3}
{'x': 1, 'y': 9}
{'x': 1, 'y': 8}
{'x': 1, 'y': 7}
{'x': 1, 'y': 6}
{'x': 1, 'y': 5}
{'x': 1, 'y': 4}
```

MP-2 Tutorial - 6

Problem 2

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

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

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

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

# (coin_value*num_of_coints) <= 60

problem.addVariable("1 rupee", range(61))
problem.addVariable("2 rupee", range(31))
problem.addVariable("5 rupee", range(13))
problem.addVariable("10 rupee", range(7))
problem.addVariable("20 rupee", range(4))

problem.addConstraint(
    constraint.ExactSumConstraint(60,[1,2,5,10,20]),["1 rupee", "2 rupee", "5 rupee","10 rupee", "2
0 rupee"])
```

In [4]:
```
def custom_constraint(a, b, c, d, e):
    if a + 2*b + 5*c + 10*d + 20*e == 60:
        return True
    problem.addConstraint(o, ["1 rupee", "2 rupee", "5 rupee","10 rupee", "20 rupee"])
```

In [5]:
```
def print_solutions(solutions):
    for s in solutions:
        print("---")
        print("""
        1 rupee: {0:d}
        2 rupee: {1:d}
        5 rupee: {2:d}
        10 rupee: {3:d}
        20 rupee: {4:d}""".format(s["1 rupee"], s["2 rupee"], s["5 rupee"], s["10 rupee"], s["20 ru
pee"]))
        print("Total:", s["1 rupee"] + s["2 rupee"]*2 + s["5 rupee"]*5 + s["10 rupee"]*10 + s["20 r
upee"]*20)
        print("---")
```

In [ ]:
```
solutions = problem.getSolutions()
print_solutions(solutions)
print("Total number of ways: {}".format(len(solutions)))
```

- Collapse

```
---

        1 rupee: 0
        2 rupee: 0
        5 rupee: 0
        10 rupee: 0
        20 rupee: 3
Total: 60
---
---

        1 rupee: 0
        2 rupee: 0
        5 rupee: 0
        10 rupee: 2
        20 rupee: 2
Total: 60
---
---

        1 rupee: 0
        2 rupee: 0
        5 rupee: 2
        10 rupee: 1
        20 rupee: 2
Total: 60
---
---

        1 rupee: 1
        2 rupee: 2
        5 rupee: 1
        10 rupee: 1
        20 rupee: 2
Total: 60
---
```

```
---

        1 rupee: 54
        2 rupee: 3
        5 rupee: 0
        10 rupee: 0
        20 rupee: 0
Total: 60
---
---

        1 rupee: 56
        2 rupee: 2
        5 rupee: 0
        10 rupee: 0
        20 rupee: 0
Total: 60
---
---

        1 rupee: 58
        2 rupee: 1
        5 rupee: 0
        10 rupee: 0
        20 rupee: 0
Total: 60
---
---

        1 rupee: 60
        2 rupee: 0
        5 rupee: 0
        10 rupee: 0
        20 rupee: 0
Total: 60
---
Total number of ways: 782
```

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-pac
kages (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'], soluti
on_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
```