Name: Vishal Ramane
Roll.No: 2019430010

Experiment No.7

Aim: Implementation of Simplex algorithm.

Problem Statement:
Suppose a company manufactures different electronic components for computers. Component-X requires 2 hours of fabrication and 1 hour of assembly; component-Y requires 3 hours of fabrication and 1 hour of assembly; component-Z requires 2 hours of fabrication and 2 hour of assembly. The company has up-to 1000 labor-hours for fabrication time and 800 labor-hours of assembly time each week. If profit on each component X,Y and Z is $7, $8, $10 resp. How many of each components should be produced to maximize the profit?

.i.e,

we can represent above problem into maximization problem as:

```
max P = 7x + 8y + 10z
        st:
        2x + 3y + 2z <= 1000
        x + y + 2z <= 800
        x,y,z >= 0
```

Objective:

- Implement a function that generates a matrix of correct size. (i.e,Simplex Tableau)
- Implement a function that check if the the current solution is optimal.
- Implement a function that determines where a pivot element is located.
- Implement a function that pivots about an element.
- Implement a function to receive string input and insert float variables into matrix.
- Implement a function to maximize and minimize the problem.

Methodology:

- Methodology of working is same as the Simplex Algorithm in CLRS Book.
- Format of the Initial Simplex-Tableau used in this implementation:

$$\begin{array}{l} \quad\ [\ P \quad X \quad Y \quad Z \quad S_1 \quad S_2 \quad \text{Total} \quad] \\ S_1\ [\ . \quad . \quad . \quad . \quad . \quad . \quad \text{cnst.val1} \quad] \\ S_2\ [\ . \quad . \quad . \quad . \quad . \quad . \quad \text{cnst.val2} \quad] \\ ..\ [\ . \quad . \quad . \quad . \quad . \quad . \quad . \quad] \end{array}$$

- Where S_1 , S_2 are Slack variables and cnst.val : represents constraint values in inequality.
- Total : represents total max. total profit achieved.
- Position of pivot at each iteration and total value at final iteration together will yield us the final results.

Name: Vishal Ramane
Roll.No: 2019430010

Experiment No.7

Implementation:

- Implement *class Tableau* with following member functions:
  - *def add_constraint(self, expression, value)*: to define and add constraint inequality and value.
  - *def _pivot_column(self)*
  - *def _pivot_row(self, col)*
  - *def _pivot(self, row, col)*: these 3 methods are used to find the position of the pivot.
  - *def _check(self)*: used to check if another iteration of the algorithm is needed.
  - *def solve(self)*: the back-bone algorithm which does computation and builds tableau.
  - *def display(self)*: used to display the tableau on terminal.

Time Complexity Analysis:

- The simplex algorithm indeed visits all $\mathbf{2^n}$ vertices in the **worst case** (“**Klee & Minty**", **1972**), and this turns out to be true for any deterministic pivot rule.

- However, in a landmark paper using a smoothed analysis, “**Spielman and Teng**”, (2001) proved that when the inputs to the algorithm are slightly randomly perturbed, the expected running time of the simplex algorithm is **polynomial** for any inputs.

- Afterwards, “**Kelner and Spielman**”,(2006) introduced **a polynomial time** randomized simplex algorithm that truly works on any inputs.

Results:

Input :

```
"""
    max P = 7x + 8y + 10z
    st:
    2x + 3y + 2z <= 1000
    x + y + 2z <= 800
    x,y,z >= 0
"""
t = Tableau([-7, -8, -10])
t.add_constraint([2, 3, 2], 1000)
t.add_constraint([1, 1, 2], 800)
t.solve()
```

Name: Vishal Ramane
Roll.No: 2019430010

Experiment No.7

Output:

PROBLEMS 100 OUTPUT TERMINAL 1: Code

```
(base) C:\Users\Vishal Ramane\OneDrive\College\AAC Lab\Code\Expt7>python -u "c:\Users\
Vishal Ramane\OneDrive\College\AAC Lab\Code\Expt7\SimplexMethod.py"

 [[   1.   -7.   -8.  -10.    0.    0.    0.]
 [   0.    2.    3.    2.    1.    0. 1000.]
 [   0.    1.    1.    2.    0.    1.  800.]]

pivot column: 4
pivot row: 3

 [[ 1.e+00 -2.e+00 -3.e+00  0.e+00  0.e+00  5.e+00  4.e+03]
 [ 0.e+00  1.e+00  2.e+00  0.e+00  1.e+00 -1.e+00  2.e+02]
 [ 0.e+00  5.e-01  5.e-01  1.e+00  0.e+00  5.e-01  4.e+02]]

pivot column: 3
pivot row: 2

 [[ 1.0e+00 -5.0e-01  0.0e+00  0.0e+00  1.5e+00  3.5e+00  4.3e+03]
 [ 0.0e+00  5.0e-01  1.0e+00  0.0e+00  5.0e-01 -5.0e-01  1.0e+02]
 [ 0.0e+00  2.5e-01  0.0e+00  1.0e+00 -2.5e-01  7.5e-01  3.5e+02]]

pivot column: 2
pivot row: 2

 [[ 1.0e+00  0.0e+00  1.0e+00  0.0e+00  2.0e+00  3.0e+00  4.4e+03]
 [ 0.0e+00  1.0e+00  2.0e+00  0.0e+00  1.0e+00 -1.0e+00  2.0e+02]
 [ 0.0e+00  0.0e+00 -5.0e-01  1.0e+00 -5.0e-01  1.0e+00  3.0e+02]]

(base) C:\Users\Vishal Ramane\OneDrive\College\AAC Lab\Code\Expt7>
```

Total Profit (row1,col7): 4.4e+03 = 4.4 $*10^3$ = 4400.

Let,

Output1 : (row2,col7)

Output2 : (row3,col7)

Finding output:

[Iterations	Pivot (row,col)	output	]
[01	at (3,4) = Z	output2	]
[02	at (2,3) = Y	output1	]
[03	at (2,2) = X	output1	]

Hence, finally.

X = Output1 : (row2,col7) = 2.0e+02 = 200,

Z = Output1 : (row2,col7) = 3.0e+02 = 300

Y = S_1= S_2= 0.

Name: Vishal Ramane
Roll.No: 2019430010

Experiment No.7

Therefore for max.profit of $4400 per week , X and Z components should be manufactured in the quantity of 200 and 300 resp.with '0' - Y components and no extra labor-hours for fabrication or assembly.(since $S_1 = S_2 = 0$).

Conclusions:

Thus we have successfully implemented Simplex algorithm.