

## ALGORITMO DE SIMPLEX Y PROBLEMAS

### 1. Algoritmo

El programa está elaborado en PYTHON 2.7

#### 1.1. Funcionamiento del programa:

1. El programa te pedirá que se ingresen las dimensiones de la matriz.
2. Luego que se ingrese cada elemento de la fila de dicha matriz.  
Nota: Al terminar de ingresar los elementos de una fila, se te pedirá que se ingrese si dicha fila corresponde a una desigualdad de la forma " $\leq$ " o simplemente a una igualdad " $=$ ".
3. EL programa te mostrará la matriz de un sistema canónico con el cual se trabajará, determinada por el adiconamiento de las variables de holgura.
4. Luego se determinará la matriz B (base) inicial con la que se empezarán los procesos indicados posteriormente en cada desarrollo de los problemas.
5. Una vez determinado la solución óptima. Se te pedirá que si deseas continuar con la determinación de la solución óptima de otro problema. De continuar, se introduce la palabra SI; de lo contrario, la palabra NO.

```
1 from math import *
2 from time import sleep
3 from numpy import *
4 from Tkinter import *
5
6 print 'A_CONTINUACION_INGRESE_LOS_DATOS_DEL_PROBLEMA_A_MINIMIZAR.'
7 print ''
8 def vuelva():
9     print 'INGRESE_LOS_DATOS_DE_OTRO_PROBLEMA:'
10    print ''
11    main()
12 def main():
13     A=[]
14     b=[]
15     cont=0
16     m=input('Ingresa el numero de filas de la matriz A:')
17     n=input('Ingresa el numero de columnas de la matriz A:')
18     for i in range(m):
19         A.append([0]*(n))
20     for i in range(m):
21         b.append(0)
22     for i in range(m):
23         print 'Ingresa los %d elementos de la fila %d:' %(n,i+1)
24         for j in range(n):
25             A[i][j]=input('')
26         simb=raw_input('Escriba " <=" o "=" segun le corresponda a esta fila:')
27         if '<='==simb:
28             cont+=1
29         for j in range (m):
30             A[j].append(0)
31         A[i][n-1+cont]=1
```

```

32     print 'Ingrese el vector b:'
33     for i in range(m):
34         b[i]=input('')
35     print 'Ingrese el vector Costo'
36     c=[]
37     c1=[]
38     for i in range(n+m):
39         c.append(0)
40         c1.append(0)
41     for i in range(n):
42         c[i]=input('')
43
44     print 'El metodo simplex trabaja ahora de la siguiente forma:'
45     print ''
46     print 'Matriz A:'
47     for i in range(m):
48         print A[i]
49     print ''
50     print 'Vector b:'
51     print b
52     print ''
53     print 'Vector Costo:'
54     print c
55     print ''
56     simplex(A,b,m,n,c,c1)
57 def combi(A,m,n):
58     B=[]
59     for i in range(m):
60         B.append([0]*(m))
61     piv=[]
62     for i in range(m):
63         piv.append(0)
64     todcom = []
65     aux = [i for i in range(m)]
66     cond = [k for k in range(n-m,n)]
67
68     todcom = todcom + [guar(aux)]
69
70     while todcom[-1]!=cond:
71         if aux[-1]!=n-1:
72             aux[-1] = aux[-1] + 1
73             todcom = todcom + [guar(aux)]
74             boolean,index = verif(todcom[-1],cond)
75             if index == 0:
76                 break
77             elif boolean == False and index!=0:
78                 aux = decen(todcom[-1],index)
79                 todcom = todcom + [guar(aux)]
80     for i in range(len(todcom)):
81         for j in range (m):
82             piv[j]=todcom[i][j]
83             for l in range (m):
84                 B[l][j]=A[l][todcom[i][j]]
85             if linalg.det(B)!=0:
86                 break
87     return B,piv
88 def guar(aux1):
89     temp = []
90     for t in range(len(aux1)):
91         temp.append(aux1[t])
92     return temp

```

```

93
94 def verif (arra,cond):
95     m = len(arra)
96     boolean = True
97     k = -1
98     for i in range(m):
99         if arra[i]==cond[i]:
100             k = i
101             boolean = False
102             break
103     return boolean,k
104
105 def decen(arra,k):
106     m = len(arra)
107     temp = []
108     for i in range(m):
109         if i == k-1:
110             temp = temp+[arra[i] + 1]
111         elif i > k-1:
112             temp = temp+[temp[i-1] + 1]
113         else:
114             temp = temp + [arra[i]]
115     return temp
116
117 def simplex(A,b,m,n,c,c1):
118     piv=[]
119     for i in range(m):
120         piv.append(0)
121     B,piv=combi(A,m,n)
122     print 'matriz_B_(inicial)_tomada:'
123     for i in range (m):
124         print B[i]
125     print ''
126     no_acotado=1
127     optimo=1
128     u=[]
129     cB=[]
130     piv2=[]
131     b1=[]
132     a1_h=[]
133     for i in range(m):
134         u.append(0)
135         cB.append(0)
136         piv2.append(0)
137         b1.append(0)
138         a1_h.append(0)
139     for i in range(m):
140         cB[i]=c[piv[i]]
141     while (optimo ==1) and (no_acotado==1):
142         Bi=[]
143         Bi=linalg.inv(B)
144         B2=[]
145         C=[]
146         for i in range(n+m):
147             C.append(0)
148         for i in range(m):
149             B2.append([0]*(m))
150         for i in range(m):
151             for j in range(m):
152                 B2[i][j]= '%.2f' %Bi[i][j]
153         print 'B^(-1),_inversa_de_B:'

```

```

154     for i in range(m):
155         print B2[i]
156     print ''
157     print 'cB:',cB
158     print ''
159     for i in range(m):
160         suma=0
161         for j in range(m):
162             suma=suma+cB[j]*Bi[j][i]
163         u[i]=suma
164     for i in range(n+m):
165         suma=0
166         for j in range(m):
167             suma=suma+u[j]*A[j][i]
168         c1[i]=suma
169     for i in range(m+n):
170         c1[i]= c[i]-c1[i]
171         if c1[i] < 10*(-14) and c1[i] > -10*(-14):
172             c1[i]=0
173     p=0
174     for i in range(n+m):
175         C[i]='%.2f' %c1[i]
176     print 'vector_de_'Costos_Reducidos':
177     print C
178     print ''
179     for j in range(n+m):
180         if c1[j]<0:
181             break
182         else: p=j
183     if p==(n+m-1):
184         optimo = 0
185         print 'Todos_los_elementos_del_Costo_Reducido_no_son_negativos,entonces:'
186         print 'LA_SOLUCION_ES_OPTIMA'
187         print ''
188     else:
189         print 'Existe_al_menos_un_elemento_del_Costo_Reducido_menor_que_cero,entonces:'
190         print 'LA_SOLUCION_AUN_NO_ES_OPTIMA'
191         print ''
192         for h in range(n+m):
193             if c1[h]<0:
194                 break
195         print 'Columna_entrante_'h'_de_la_matriz_A_a_la_matriz_B:',h+1
196         print ''
197         for i in range(m):
198             suma=0
199             for j in range(m):
200                 suma=suma+Bi[i][j]*b[j]
201             b1[i]=suma
202         b2=[]
203         for i in range (m):
204             b2.append(0)
205             b2[i]='%.2f' %b1[i]
206         print 'Calculando_b1=(B^(-1))*b:'
207         print b2
208         print ''
209         for i in range(m):
210             suma=0
211             for j in range(m):
212                 suma=suma+Bi[i][j]*A[j][h]
213             a1_h[i]=suma
214         b2=[]

```

```

215         for i in range(m):
216             b2.append(0)
217             b2[i]='%.2f' %a1_h[i]
218         print 'Calculando(B^(-1))*a_( %d),_siendo_a_( %d)_la_columna_ %d_de_a' %(h+1,
219 h+1,h+1)
220         print b2
221         print ''
222         p=0
223         for i in range(m):
224             if a1_h[i]>0:
225                 break
226             else: p=i
227         if p==(m-1):
228             print 'Todos_los_elementos_del_resultado_anterior_no_son_positivos,_entonces:'
229             print 'LA_SOLUCION_NO_ESTA_ACOTADA'
230             print ''
231             no_acotado=0
232         else:
233
234             for i in range(m):
235                 if a1_h[i]>0:
236                     minimo= b1[i]*(a1_h[i]**(-1))
237                     r=i
238                     break
239             for j in range(i+1,m):
240                 if a1_h[j]>0:
241                     if (b1[j]*(a1_h[j]**(-1))) < minimo:
242                         minimo = b1[j]*(a1_h[j]**(-1))
243                         r=j
244             print 'Columna_saliente_”r”_de_la_matriz_B:',r+1
245             print ''
246             for i in range(m):
247                 B[i][r]=A[i][h]
248             print 'Haciendo_el_cambio_de_la_columna_ %d_de_B_con_la_columna_ %d_de_A'
249 %(r+1,h+1)
250             print 'Matriz_B_mejorada:'
251             for i in range(m):
252                 print B[i]
253             print ''
254             cB[r]=c[h]
255             piv[r]=h
256             for i in range(m):
257                 piv2[i]=piv[i]+1
258
259         if optimo ==0:
260             for i in range(m):
261                 suma=0
262                 for j in range(m):
263                     suma=suma+B[i][j]*b[j]
264                 b1[i]=suma
265             x=[]
266             for i in range(m+n):
267                 x.append(0)
268             for i in range(m):
269                 x[piv[i]]='%.3f' %b1[i]
270             print 'Solucion:'
271             print x[0:n]
272             print ''
273         conf=raw_input ('Diga_SI_o_NO_si_desea_continuar_con_mas_problemas:')
274         if 'SI' == conf:
275             vuelva()

```

```

276     else:
277         print 'LA_TAREA_HA_FINALIZADO'
278 main()

```

### Observación 1:

Para el desarrollo de este algoritmo se utiliza la biblioteca numpy.

### Observación 2:

Se puede trabajar ya sea con un sistema estándar o canónico, debido a que el programa te pedirá que especifiques si existe una desigualdad o igualdad en cada paso.

## 1.2. Problema 1:

Obtener la solución óptima que minimice la siguiente función:

$$f(x_1, x_2) = -x_1 - 2x_2,$$

teniendo las siguientes condiciones:

$$\begin{array}{rcl}
 2x_1 & + & 6x_2 \leq 15 \\
 28x_1 & + & 8x_2 \leq 77 \\
 & & x_1 \geq 0 \\
 & & x_2 \geq 0
 \end{array}$$

El algoritmo nos da como solución a nuestro problema:

$$[2, 250; 1, 750],$$

mostrando los siguientes datos obtenidos en el proceso.

```

1  >>>
2  A CONTINUACION INGRESE LOS DATOS DEL PROBLEMA A MINIMIZAR.
3
4  Ingrese el numero de filas de la matriz A:2
5  Ingrese el numero de columnas de la matriz A:2
6  Ingrese los 2 elementos de la fila 1:
7  2
8  6
9  Escriba "<=" o ">=" segun le corresponda a esta fila:<=
10 Ingrese los 2 elementos de la fila 2:
11 28
12 8
13 Escriba "<=" o ">=" segun le corresponda a esta fila:<=
14 Ingrese el vector b:
15 15
16 77
17 Ingrese el vector Costo
18 -1
19 -2
20 El metodo simplex trabaja ahora de la siguiente forma:
21
22 Matriz A:
23 [2, 6, 1, 0]
24 [28, 8, 0, 1]
25
26 Vector b:
27 [15, 77]
28
29 Vector Costo:
30 [-1, -2, 0, 0]
31
32 matriz B ( inicial ) tomada:
33 [2, 6]

```

```

34 [28, 8]
35
36 B^(-1), inversa de B:
37 [-0.05, 0.04]
38 [0.18, -0.01]
39
40 cB: [-1, -2]
41
42 vector de "Costos_Reducidos":
43 [0.00, 0.00, 0.32, 0.01]
44
45 Todos los elementos del Costo Reducido no son negativos, entonces:
46 LA SOLUCION ES OPTIMA
47
48 Solucion:
49 [2.250, 1.750]

```

### 1.3. Problema 2:

Obtener la solución óptima que minimice la siguiente función:

$$f(x_1, x_2, x_3) = x_1 + x_2 - 4x_3$$

teniendo las siguientes condiciones:

$$\begin{array}{rclcl}
 x_1 & + & x_2 & + & 2x_3 & \leq & 9 \\
 x_1 & + & x_2 & - & x_3 & \leq & 2 \\
 -x_1 & + & x_2 & + & x_3 & \leq & 4 \\
 & & & & x_1 & \geq & 0 \\
 & & & & x_2 & \geq & 0 \\
 & & & & x_3 & \geq & 0
 \end{array}$$

El algoritmo nos da como solución a nuestro problema:

$$[0,333; 0; 4,333]$$

mostrando los siguientes datos obtenidos en el proceso.

```

1 Diga SI o NO si desea continuar con mas problemas:SI
2 INGRESE LOS DATOS DE OTRO PROBLEMA:
3
4 Ingrese el numero de filas de la matriz A:3
5 Ingrese el numero de columnas de la matriz A:3
6 Ingrese los 3 elementos de la fila 1:
7 1
8 1
9 2
10 Escriba "<=" o "=" segun le corresponda a esta fila:<=
11 Ingrese los 3 elementos de la fila 2:
12 1
13 1
14 -1
15 Escriba "<=" o "=" segun le corresponda a esta fila:<=
16 Ingrese los 3 elementos de la fila 3:
17 -1
18 1
19 1
20 Escriba "<=" o "=" segun le corresponda a esta fila:<=
21 Ingrese el vector b:
22 9
23 2
24 4
25 Ingrese el vector Costo

```

```

26 | 1
27 | 1
28 | -4
29 | El metodo simplex trabaja ahora de la siguiente forma:
30 |
31 | Matriz A:
32 | [1, 1, 2, 1, 0, 0]
33 | [1, 1, -1, 0, 1, 0]
34 | [-1, 1, 1, 0, 0, 1]
35 |
36 | Vector b:
37 | [9, 2, 4]
38 |
39 | Vector Costo:
40 | [1, 1, -4, 0, 0, 0]
41 |
42 | matriz B ( inicial ) tomada:
43 | [1, 1, 2]
44 | [1, 1, -1]
45 | [-1, 1, 1]
46 |
47 |  $B^{-1}$ , inversa de B:
48 | ['0.33', '0.17', '-0.50']
49 | ['0.00', '0.50', '0.50']
50 | ['0.33', '-0.33', '-0.00']
51 |
52 | cB: [1, 1, -4]
53 |
54 | vector de "Costos_Reducidos":
55 | ['0.00', '0.00', '0.00', '1.00', '-2.00', '0.00']
56 |
57 | Existe al menos un elemento del Costo Reducido menor que cero, entonces:
58 | LA SOLUCION AUN NO ES OPTIMA
59 |
60 | Columna entrante "h" de la matriz A a la matriz B: 5
61 |
62 | Calculando  $b_1 = (B^{-1}) * b$ :
63 | ['1.33', '3.00', '2.33']
64 |
65 | Calculando  $(B^{-1}) * a_5$ , siendo  $a_5$  la columna 5 de a
66 | ['0.17', '0.50', '-0.33']
67 |
68 | Columna saliente "r" de la matriz B: 2
69 |
70 | Haciendo el cambio de la columna 2 de B con la columna 5 de A
71 | Matriz B mejorada:
72 | [1, 0, 2]
73 | [1, 1, -1]
74 | [-1, 0, 1]
75 |
76 |  $B^{-1}$ , inversa de B:
77 | ['0.33', '0.00', '-0.67']
78 | ['0.00', '1.00', '1.00']
79 | ['0.33', '0.00', '0.33']
80 |
81 | cB: [1, 0, -4]
82 |
83 | vector de "Costos_Reducidos":
84 | ['0.00', '4.00', '0.00', '1.00', '0.00', '2.00']
85 |
86 | Todos los elementos del Costo Reducido no son negativos, entonces:

```



```

87 LA SOLUCION ES OPTIMA
88
89 Solucion:
90 ['0.333', 0, '4.333']

```

### 1.4. Problema 3:

En este problema, tendremos que minimizar la siguiente función:

$$f(x_1, x_2, x_3, x_4, x_5) = 7x_1 + 6x_2 + 5x_3 - 2x_4 + 3x_5,$$

teniendo las siguientes condiciones:

$$\begin{array}{rclclclclcl}
 x_1 & + & 3x_2 & + & 5x_3 & - & 2x_4 & + & 2x_5 & \leq & 4 \\
 2x_1 & + & 4x_2 & + & 4x_3 & - & 2x_4 & + & 5x_5 & \leq & 5 \\
 3x_1 & + & x_2 & + & 2x_3 & - & x_4 & - & 2x_5 & \leq & 1 \\
 & & & & & & & & x_1 & \geq & 0 \\
 & & & & & & & & x_2 & \geq & 0 \\
 & & & & & & & & x_3 & \geq & 0 \\
 & & & & & & & & x_4 & \geq & 0 \\
 & & & & & & & & x_5 & \geq & 0
 \end{array}$$

El algoritmo nos dice que nuestra solución no está acotada, mostrando los siguientes datos obtenidos en el proceso.

```

1 Diga SI o NO si desea continuar con mas problemas:SI
2 INGRESE LOS DATOS DE OTRO PROBLEMA:
3
4 Ingrese el numero de filas de la matriz A:3
5 Ingrese el numero de columnas de la matriz A:5
6 Ingrese los 5 elementos de la fila 1:
7 1
8 3
9 5
10 -2
11 2
12 Escriba "≤" o "=" segun le corresponda a esta fila:≤
13 Ingrese los 5 elementos de la fila 2:
14 2
15 4
16 4
17 -2
18 5
19 Escriba "≤" o "=" segun le corresponda a esta fila:≤
20 Ingrese los 5 elementos de la fila 3:
21 3
22 1
23 2
24 -1
25 -2
26 Escriba "≤" o "=" segun le corresponda a esta fila:≤
27 Ingrese el vector b:
28 4
29 5
30 1
31 Ingrese el vector Costo
32 7
33 6
34 5
35 -2
36 3
37 El metodo simplex trabaja ahora de la siguiente forma:

```

```

38
39 Matriz A:
40 [1, 3, 5, -2, 2, 1, 0, 0]
41 [2, 4, 4, -2, 5, 0, 1, 0]
42 [3, 1, 2, -1, -2, 0, 0, 1]
43
44 Vector b:
45 [4, 5, 1]
46
47 Vector Costo:
48 [7, 6, 5, -2, 3, 0, 0, 0]
49
50 matriz B ( inicial ) tomada:
51 [1, 3, 5]
52 [2, 4, 4]
53 [3, 1, 2]
54
55  $B^{-1}$ , inversa de B:
56 ['-0.18', '0.05', '0.36']
57 ['-0.36', '0.59', '-0.27']
58 ['0.45', '-0.36', '0.09']
59
60 cB: [7, 6, 5]
61
62 vector de "Costos.Reducidos":
63 ['0.00', '0.00', '0.00', '1.09', '-2.14', '1.18', '-2.05', '-1.36']
64
65 Existe al menos un elemento del Costo Reducido menor que cero, entonces:
66 LA SOLUCION AUN NO ES OPTIMA
67
68 Columna entrante "h" de la matriz A a la matriz B: 5
69
70 Calculando  $b_1 = (B^{-1}) * b$ :
71 ['-0.14', '1.23', '0.09']
72
73 Calculando  $(B^{-1}) * a_5$ , siendo  $a_5$  la columna 5 de a
74 ['-0.86', '2.77', '-1.09']
75
76 Columna saliente "r" de la matriz B: 2
77
78 Haciendo el cambio de la columna 2 de B con la columna 5 de A
79 Matriz B mejorada:
80 [1, 2, 5]
81 [2, 5, 4]
82 [3, -2, 2]
83
84  $B^{-1}$ , inversa de B:
85 ['-0.30', '0.23', '0.28']
86 ['-0.13', '0.21', '-0.10']
87 ['0.31', '-0.13', '-0.02']
88
89 cB: [7, 3, 5]
90
91 vector de "Costos.Reducidos":
92 ['0.00', '0.77', '0.00', '0.95', '0.00', '0.90', '-1.59', '-1.57']
93
94 Existe al menos un elemento del Costo Reducido menor que cero, entonces:
95 LA SOLUCION AUN NO ES OPTIMA
96
97 Columna entrante "h" de la matriz A a la matriz B: 7
98

```

```

99 | Calculando  $b1=(B^{(-1)})*b$ :
100 | ['0.25', '0.44', '0.57']
101 |
102 | Calculando  $(B^{(-1)})*a_{(7)}$ , siendo  $a_{(7)}$  la columna 7 de a
103 | ['0.23', '0.21', '-0.13']
104 |
105 | Columna saliente "r" de la matriz B: 1
106 |
107 | Haciendo el cambio de la columna 1 de B con la columna 7 de A
108 | Matriz B mejorada:
109 | [0, 2, 5]
110 | [1, 5, 4]
111 | [0, -2, 2]
112 |
113 |  $B^{(-1)}$ , inversa de B:
114 | ['-1.29', '1.00', '1.21']
115 | ['0.14', '0.00', '-0.36']
116 | ['0.14', '0.00', '0.14']
117 |
118 | cB: [0, 3, 5]
119 |
120 | vector de "Costos_Reducidos":
121 | ['6.93', '2.93', '0.00', '-0.07', '0.00', '-1.14', '0.00', '0.36']
122 |
123 | Existe al menos un elemento del Costo Reducido menor que cero, entonces:
124 | LA SOLUCION AUN NO ES OPTIMA
125 |
126 | Columna entrante "h" de la matriz A a la matriz B: 4
127 |
128 | Calculando  $b1=(B^{(-1)})*b$ :
129 | ['1.07', '0.21', '0.71']
130 |
131 | Calculando  $(B^{(-1)})*a_{(4)}$ , siendo  $a_{(4)}$  la columna 4 de a
132 | ['-0.64', '0.07', '-0.43']
133 |
134 | Columna saliente "r" de la matriz B: 2
135 |
136 | Haciendo el cambio de la columna 2 de B con la columna 4 de A
137 | Matriz B mejorada:
138 | [0, -2, 5]
139 | [1, -2, 4]
140 | [0, -1, 2]
141 |
142 |  $B^{(-1)}$ , inversa de B:
143 | ['0.00', '1.00', '-2.00']
144 | ['2.00', '-0.00', '-5.00']
145 | ['1.00', '-0.00', '-2.00']
146 |
147 | cB: [0, -2, 5]
148 |
149 | vector de "Costos_Reducidos":
150 | ['6.00', '3.00', '0.00', '0.00', '1.00', '-1.00', '0.00', '0.00']
151 |
152 | Existe al menos un elemento del Costo Reducido menor que cero, entonces:
153 | LA SOLUCION AUN NO ES OPTIMA
154 |
155 | Columna entrante "h" de la matriz A a la matriz B: 6
156 |
157 | Calculando  $b1=(B^{(-1)})*b$ :
158 | ['3.00', '3.00', '2.00']
159 |

```

```

160 Calculando(B^(-1))*a_(6), siendo a_(6) la columna 6 de a
161 ['0.00', '2.00', '1.00']
162
163 Columna saliente "r" de la matriz B: 2
164
165 Haciendo el cambio de la columna 2 de B con la columna 6 de A
166 Matriz B mejorada:
167 [0, 1, 5]
168 [1, 0, 4]
169 [0, 0, 2]
170
171 B^(-1), inversa de B:
172 ['0.00', '1.00', '-2.00']
173 ['1.00', '0.00', '-2.50']
174 ['0.00', '0.00', '0.50']
175
176 cB: [0, 0, 5]
177
178 vector de "Costos.Reducidos":
179 ['-0.50', '3.50', '0.00', '0.50', '8.00', '0.00', '0.00', '-2.50']
180
181 Existe al menos un elemento del Costo Reducido menor que cero, entonces:
182 LA SOLUCION AUN NO ES OPTIMA
183
184 Columna entrante "h" de la matriz A a la matriz B: 1
185
186 Calculando b1=(B^(-1))*b:
187 ['3.00', '1.50', '0.50']
188
189 Calculando(B^(-1))*a_(1), siendo a_(1) la columna 1 de a
190 ['-4.00', '-6.50', '1.50']
191
192 Columna saliente "r" de la matriz B: 3
193
194 Haciendo el cambio de la columna 3 de B con la columna 1 de A
195 Matriz B mejorada:
196 [0, 1, 1]
197 [1, 0, 2]
198 [0, 0, 3]
199
200 B^(-1), inversa de B:
201 ['0.00', '1.00', '-0.67']
202 ['1.00', '0.00', '-0.33']
203 ['0.00', '0.00', '0.33']
204
205 cB: [0, 0, 7]
206
207 vector de "Costos.Reducidos":
208 ['0.00', '3.67', '0.33', '0.33', '7.67', '0.00', '0.00', '-2.33']
209
210 Existe al menos un elemento del Costo Reducido menor que cero, entonces:
211 LA SOLUCION AUN NO ES OPTIMA
212
213 Columna entrante "h" de la matriz A a la matriz B: 8
214
215 Calculando b1=(B^(-1))*b:
216 ['4.33', '3.67', '0.33']
217
218 Calculando(B^(-1))*a_(8), siendo a_(8) la columna 8 de a
219 ['-0.67', '-0.33', '0.33']
220

```

```

221 Columna saliente "r" de la matriz B: 3
222
223 Haciendo el cambio de la columna 3 de B con la columna 8 de A
224 Matriz B mejorada:
225 [0, 1, 0]
226 [1, 0, 0]
227 [0, 0, 1]
228
229 B(-1), inversa de B:
230 ['0.00', '1.00', '0.00']
231 ['1.00', '0.00', '0.00']
232 ['0.00', '0.00', '1.00']
233
234 cB: [0, 0, 0]
235
236 vector de "Costos_Reducidos":
237 ['7.00', '6.00', '5.00', '-2.00', '3.00', '0.00', '0.00', '0.00']
238
239 Existe al menos un elemento del Costo Reducido menor que cero, entonces:
240 LA SOLUCION AUN NO ES OPTIMA
241
242 Columna entrante "h" de la matriz A a la matriz B: 4
243
244 Calculando b1=(B(-1))*b:
245 ['5.00', '4.00', '1.00']
246
247 Calculando(B(-1))*a(4), siendo a(4) la columna 4 de a
248 ['-2.00', '-2.00', '-1.00']
249
250 Todos los elementos del resultado anterior no son positivos, entonces:
251 LA SOLUCION NO ESTA ACOTADA

```

### 1.5. Problema 4:

Obtener la solución óptima que minimice la siguiente función:

$$f(x_1, x_2) = -2x_1 - x_2,$$

teniendo las siguientes condiciones:

$$\begin{array}{rcl}
 x_1 + x_2 & \leq & 2 \\
 x_1 & \leq & 1 \\
 x_1 & \geq & 0 \\
 x_2 & \geq & 0
 \end{array}$$

El algoritmo nos da como solución a nuestro problema:

$$[1,000; 1,000]$$

mostrando los siguientes datos obtenidos en el proceso.

```

1 Diga SI o NO si desea continuar con mas problemas:SI
2 INGRESE LOS DATOS DE OTRO PROBLEMA:
3
4 Ingrese el numero de filas de la matriz A:2
5 Ingrese el numero de columnas de la matriz A:2
6 Ingrese los 2 elementos de la fila 1:
7 1
8 1
9 Escriba "≤" o "=" segun le corresponda a esta fila:≤
10 Ingrese los 2 elementos de la fila 2:
11 1
12 0
13 Escriba "≤" o "=" segun le corresponda a esta fila:≤

```

```

14 | Ingrese el vector b:
15 | 2
16 | 1
17 | Ingrese el vector Costo
18 | -2
19 | -1
20 | El metodo simplex trabaja ahora de la siguiente forma:
21 |
22 | Matriz A:
23 | [1, 1, 1, 0]
24 | [1, 0, 0, 1]
25 |
26 | Vector b:
27 | [2, 1]
28 |
29 | Vector Costo:
30 | [-2, -1, 0, 0]
31 |
32 | matriz B ( inicial ) tomada:
33 | [1, 1]
34 | [1, 0]
35 |
36 | B^(-1), inversa de B:
37 | ['0.00', '1.00']
38 | ['1.00', '-1.00']
39 |
40 | cB: [-2, -1]
41 |
42 | vector de "Costos Reducidos":
43 | ['0.00', '0.00', '1.00', '1.00']
44 |
45 | Todos los elementos del Costo Reducido no son negativos, entonces:
46 | LA SOLUCION ES OPTIMA
47 |
48 | Solucion:
49 | ['1.000', '1.000']

```

### 1.6. Problema 5:

Obtener la solución óptima que minimice la siguiente función:

$$f(x_1, x_2, x_3) = -x_1 - 4x_2 + 3x_3,$$

teniendo las siguientes condiciones:

$$\begin{array}{rclclcl}
 x_1 & + & x_2 & + & x_3 & \leq & 6 \\
 -2x_1 & + & 3x_2 & + & 3x_3 & \leq & 3 \\
 & & x_1 & \geq & 0 & & \\
 & & x_2 & \geq & 0 & & \\
 & & x_3 & \geq & 0 & & 
 \end{array}$$

El algoritmo nos da como solución a nuestro problema:

$$[3,000; 3,000, 0],$$

mostrando los siguientes datos obtenidos en el proceso.

```

1 | Diga SI o NO si desea continuar con mas problemas:SI
2 | INGRESE LOS DATOS DE OTRO PROBLEMA:
3 |
4 | Ingrese el numero de filas de la matriz A:2
5 | Ingrese el numero de columnas de la matriz A:3
6 | Ingrese los 3 elementos de la fila 1:

```

```

7 | 1
8 | 1
9 | 1
10 | Escriba "≤" o "=" segun le corresponda a esta fila:≤=
11 | Ingrese los 3 elementos de la fila 2:
12 | -2
13 | 3
14 | 3
15 | Escriba "≤" o "=" segun le corresponda a esta fila:≤=
16 | Ingrese el vector b:
17 | 6
18 | 3
19 | Ingrese el vector Costo
20 | -1
21 | -4
22 | 3
23 | El metodo simplex trabaja ahora de la siguiente forma:
24 |
25 | Matriz A:
26 | [1, 1, 1, 1, 0]
27 | [-2, 3, 3, 0, 1]
28 |
29 | Vector b:
30 | [6, 3]
31 |
32 | Vector Costo:
33 | [-1, -4, 3, 0, 0]
34 |
35 | matriz B ( inicial ) tomada:
36 | [1, 1]
37 | [-2, 3]
38 |
39 | B-1, inversa de B:
40 | ['0.60', '-0.20']
41 | ['0.40', '0.20']
42 |
43 | cB: [-1, -4]
44 |
45 | vector de "Costos Reducidos":
46 | ['0.00', '0.00', '7.00', '2.20', '0.60']
47 |
48 | Todos los elementos del Costo Reducido no son negativos, entonces:
49 | LA SOLUCION ES OPTIMA
50 |
51 | Solucion:
52 | ['3.000', '3.000', 0]

```

### 1.7. Problema 6:

Obtener la solución óptima que minimice la siguiente función:

$$f(x_1, x_2) = -x_1 - 3x_2$$

teniendo las siguientes condiciones:

$$\begin{array}{rcl}
 2x_1 & + & 3x_2 \leq 6 \\
 -x_1 & + & x_2 \leq 1 \\
 & & x_1 \geq 0 \\
 & & x_2 \geq 0
 \end{array}$$

El algoritmo nos da como solución a nuestro problema:

$$[0,600; 1,600]$$

mostrando los siguientes datos obtenidos en el proceso.

```
1  Diga SI o NO si desea continuar con mas problemas:SI
2  INGRESE LOS DATOS DE OTRO PROBLEMA:
3
4  Ingrese el numero de filas de la matriz A:2
5  Ingrese el numero de columnas de la matriz A:2
6  Ingrese los 2 elementos de la fila 1:
7  2
8  3
9  Escriba "≤" o "=" segun le corresponda a esta fila:≤
10 Ingrese los 2 elementos de la fila 2:
11 -1
12 1
13 Escriba "≤" o "=" segun le corresponda a esta fila:≤
14 Ingrese el vector b:
15 6
16 1
17 Ingrese el vector Costo
18 -1
19 -3
20 El metodo simplex trabaja ahora de la siguiente forma:
21
22 Matriz A:
23 [2, 3, 1, 0]
24 [-1, 1, 0, 1]
25
26 Vector b:
27 [6, 1]
28
29 Vector Costo:
30 [-1, -3, 0, 0]
31
32 matriz B ( inicial ) tomada:
33 [2, 3]
34 [-1, 1]
35
36  $B^{-1}$ , inversa de B:
37 ['0.20', '-0.60']
38 ['0.20', '0.40']
39
40 cB: [-1, -3]
41
42 vector de "Costos Reducidos":
43 ['0.00', '0.00', '0.80', '0.60']
44
45 Todos los elementos del Costo Reducido no son negativos, entonces:
46 LA SOLUCION ES OPTIMA
47
48 Solucion:
49 ['0.600', '1.600']
50
51 Diga SI o NO si desea continuar con mas problemas:NO
52 LA TAREA HA FINALIZADO
```