PROGRAMACIÓN LINEAL
PROYECTO I

Condiciones para entregar el proyecto

- Equipos de dos o tres integrantes.
- Fecha: Domingo 17 de Noviembre a las 23:55 en comunidad itam.

  Cada grupo debe entregar programas y *scripts en MatLab* que permitan reproducir los resultados reportados en su documento. Entre esos deben estar:

  1. La función `mSimplexFaseII.m`
  2. La función `mSimplexDual.m`

  - Entregar todo lo siguiente en un archivo `zip`.
    - `mSimplexFaseII.m` y `generaKleeMinty.m`.
    - `testFaseII.m` que verifica con dos ejemplos exactos la implementación de `mSimplexFaseII.m`.
    - `SimplexKleeMinty.m` que genera la tabla usando la función `generaKleeMinty.m`.
    - a) algún *script* que muestre el resultado. En la documentación hay que describir el problema e interpretar los resultados.
    - La función `mSimplexDual.m`
    - `testSimplexDual.m` que verifica con un ejemplo exacto la implementación de `mSimplexDual.m`.
  - Entregar una documentación en PDF con:
    - nombres y claves únicas de los integrantes de equipo.
    - su interpretación de los resultados, en el caso de Kleen Minty.
    - sus resultados de sus ejemplos exactos, que verifiquen la implementación de `mSimplexFaseII.m` y `mSimplexDual.m`

Nota: En el sentido de *paired programming*, les recomiendo que a lo más dos personas trabajen en un archivo. Esa debe ayudarles terminar el proyecto más rápido. Así, 1 o 2 personas pueden dedicarse a algo, mientras el resto del equipo avanza en otra parte. Eso es común en equipos de "*agile development*".

1. Implementa la Fase II **revisada** para un problema del tipo

$$\begin{aligned} \min \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{sujeto a} \quad & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\,. \end{aligned} \tag{1}$$

Antes de entrar al ciclo de Fase II, la función debe introducir variables de holgura. *Pista: Recuerde cual seria la primera SBF (fácil en este caso).*

Es importante mantener los conjuntos de índices básicas y no básicas B y N.

```
function [xo, zo, ban, iter] = mSimplexFaseII(A, b, c)
% purpose: Versión del Simplex en la Fase II
%   minimizar   c^T x
%   sujeto a    Ax <= b ,   x >= 0 ,   b >= 0
%
% In : A ... mxn matrix
%      b ... column vector with as many rows as A
%      c ... column vector with as many entries as one row of A
%
% Out: xo   ... SFB óptima del problema
%      zo   ... valor óptimo del problema
%      ban  ... indica casos:
%             -1  ... si el conjunto factible es vacio
%              0  ... si se encontro una solución óptima
%              1  ... si la función objectivo no es acotada.
%      iter ... es el numeró de iteraciones (cambios de variables basicas)
%               que hizo el método
%

   ...
end
```

Tarea: Verifiquen su código con dos ejemplos de clase que se hayan resuelto exactamente.

Sugerencias:

2. El ejemplo de Klee–Minty (simplificado, tomado de Kitahara, Mizuno).
La Fase II revisada realiza $(2^m - 1)$ iteraciones para resolver el siguiente problema:

$$\begin{aligned} \text{minimizar} \quad & -\textstyle\sum_{i=1}^{m} x_i \\ \text{sujeto a} \quad & x_1 \leq 1 \\ & 2\textstyle\sum_{j=1}^{i-1} x_j + x_i \leq 2^i - 1\,,\ i = 2, \ldots, m \\ & x_i \geq 0\,,\ i = 1, \ldots, m\,. \end{aligned}$$

Resolver los casos $m = 3, 4, 5, 6, 7, 8, 9, 10$, y reportar el número de iteraciones en cada caso junto con el tiempo de máquina (en MatLab el tiempo se mide con: `tic` .. code .. `toc`).

m	número de iteraciones	cpu time

Sugerencias:

- Escribe una función `generaKleeMinty.m`.
- Después, llama esa y la función del apartado 1 en un script `SimplexKleeMinty.m` en el cual se genera la tabla.
- Te podrían servir los comandos: `ones, eye, tril`.

3. **El método Simplex dual**

Se les pide, implementar el método Simplex Dual **revisado** para un problema del tipo

$$\text{(P)} \qquad \begin{cases} \min & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{sujeto a} & A\boldsymbol{x} \geq \boldsymbol{b}\,, \quad \boldsymbol{x} \geq \boldsymbol{0}\,, \quad \boldsymbol{c} \geq \boldsymbol{0}\,. \end{cases}$$

Actividad **(2)**, *justificaciones antes de implementar:*

En la clase no hemos definido el simplex Dual para ese problema. Para facilitarles el trabajo, deben responder las siguientes afirmaciones antes de implementar. Al restar variables de holgura obtienen el problema (el cual se puede resolver con lo que vimos en clase)

$$(P_h) \qquad \begin{cases} \min & \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{0}^\top \boldsymbol{y} \\ \text{sujeto a} & A\boldsymbol{x} - \boldsymbol{y} = \boldsymbol{b}\,, \quad \boldsymbol{x}, \boldsymbol{y} \geq \boldsymbol{0}\,, \quad \boldsymbol{c} \geq \boldsymbol{0}\,. \end{cases}$$

Muestre que

a) $\boldsymbol{x} \in C_F(P) \implies \exists \boldsymbol{y} \geq \boldsymbol{0} \colon (\boldsymbol{x}, \boldsymbol{y}) \in C_F(P_h)\,,$

b) $(\boldsymbol{x}, \boldsymbol{y}) \in C_F(P_h) \implies \boldsymbol{x} \in C_F(P)\,,$

c) El dual de (P) coincide con el dual de (P_h). En particular $C_F(D) = C_F(D_h)$.

```
function [xo, zo, ban, iter, lamo] = mSimplexDual(A, b, c)
% purpose: Versión del Simplex Dual (revisado)
%   minimizar   c^T x
%   sujeto a    Ax >= b ,   x >= 0 ,   c >= 0
%
% In : A ... m x n matrix
%      b ... column vector with as many rows as A
%      c ... column vector with as many columns as A
%
% Out: xo   ... SFB óptima del problema
%      zo   ... valor óptimo del problema
%      ban  ... indica casos:
%             -1  ... si el conjunto factible es vacio
%              0  ... si se encontro una solución óptima
%              1  ... si la función objectivo no es acotada.
%      iter ... es el numeró de iteraciones (cambios de variables basicas)
%               que hizo el método
%      lamo ... Solución del problema dual
%

   ...
end
```