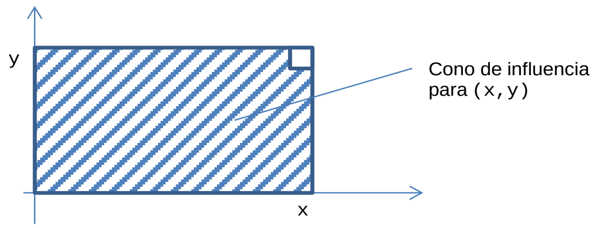
ISIS 2103 Diseño de algoritmos **Nombre:** ______________________________
Semestre 2012-10 Parcial 3 **Sección: 3**
Mayo 9, 2012
Prof. Rodrigo Cardoso

1 [45 puntos]

En un experimento de microbiología se ha desarrollado una técnica para cultivar bacterias sobre una cuadrícula de `M×N` celdas, `M,N>0`, identificadas con posiciones en `0..M×0..N`. El *cono de influencia* para una celda `(x,y)` es el conjunto de celdas `C(x,y) = (0..x × 0..y)\{(x,y)}`.



El experimento consigue que la población de bacterias alcance un estado estable en el que, llamando `b(x,y)` la población de bacterias:

(i) `b(0,0) = 1`
(ii) Para `(x,y)≠(0,0), b(x,y)` es la suma de todas las poblaciones en su cono de influencia.

Se quiere desarrollar un algoritmo de programación dinámica para calcular la población más grande `b(x,y)` en estado estable, considerando todas las celdas de la cuadrícula `0..M×0..N`. Explique su método y estime las complejidades espacial y temporal (use sumas como operaciones básicas), siguiendo los lineamientos de desarrollo que se usaron en clase (lenguaje, recurrencia, …).

N.B: Se espera una solución eficiente en tiempo y en espacio. Por tanto, soluciones menos eficientes que lo esperado pueden ser penalizadas.

Ayuda: `b(x,y)` tiene una definición especial para las posiciones `(0,0)`, `(1,1) y (0,y)`, `(x,0)` con `x,y>0`. En otros casos se puede dar una regla general sencilla, en términos de las poblaciones de las celdas vecinas a `(x,y)`.

2 [25 puntos]

ComplexPro es una empresa que se dedica al diseño y fabricación de productos complejos. El departamento de diseño de ComplexPro establece un diseño para cada producto que se pretende fabricar, definiendo un par `(E,D)`, donde

- `E` es un conjunto finito de *etapas* de producción, y
- `D:E` $\rightarrow$ `2`^{E} es una función que define, para cada etapa e, sus *dependencias,* i.e., un conjunto de etapas que dependen o necesitan que e se realice.

En un diseño `(E,D)`, una etapa e es *realizable* si todas las etapas de las que e depende ya han sido realizadas. Un producto es *factible* si hay una manera de realizar las etapas de `E` cumpliendo las restricciones de dependencia que establece `D`.

2a Diseñe un algoritmo para resolver eficientemente el problema de determinar, para un diseño `(E,D)`, si el producto diseñado es factible y, en caso de serlo, determinar cómo puede fabricarse (dar un orden de ejecución de las etapas de `E` que respete las dependencias de `D`).

2b Estime las complejidades temporal y espacial de su algoritmo.

3 [30 puntos]

Suponga un juego que consiste en manipular una matriz `a[1..m,1..n]` que contiene los números `1..m*n` sin repeticiones. La figura ilustra una posible matriz `a`, cuando `m=2`, `n=3`:

4	2	5
6	1	3

El juego permite efectuar las siguientes jugadas, para cambiar la matriz `a` a una matriz `b`:

- `b` es igual a `a`, excepto que la fila `i` ha rotado una posición a la derecha, módulo `n`, para `1≤i≤m`. Por ejemplo, a partir de la matriz `a` de arriba, se puede mover la fila `2` para llegar a

4	2	5
3	6	1

- `b` es igual a `a`, excepto que la columna `j` ha rotado una posición hacia abajo, módulo `n`, para `1≤j≤n`. Por ejemplo, a partir de la matriz `a` de arriba, se puede mover la columna `1` para llegar a

6	2	5
4	1	3

Se quiere determinar si, a partir de una matriz dada `a`, se puede producir un arreglo `b` que contenga los elementos de `1..m*n` ordenados ascendentemente, por filas y columnas. En el caso del ejemplo, se pregunta si se podría llegar a

1	2	3
4	5	6

3a Exprese el problema como una búsqueda en grafos.

3b Justifique si (i) hay que marcar nodos (ii) hay que verificar que la agenda se vacíe (iii) el algoritmo puede no terminar.

3c Estime la complejidad temporal de un algoritmo de agenda en términos de `m` y `n`.