Universidad Católica del Uruguay
Facultad de Ingeniería y Tecnologías

PROGRAMACIÓN I – SEM 1 - 2012

Trabajo Obligatorio III

Día de Entrega**: Viernes, Julio 6 de 2012**

Descripción*:*

El objetivo de este trabajo es simular el desarrollo de una epidemia sobre un tejido biológico, cuya evolución será dependiente de varios parámetros.

Para simular el tejido utilizaremos una ***grilla*** de ***N* por *N* puntos**, cada una de cuyas celdas representa:

- Una célula sana del tejido – Que será representada por un nro. 0 (cero)
- Una célula del tejido enferma – Que se representará mediante un número mayor a 0, que indica la cantidad de días que hace que está contagiada
- Un espacio vacío dentro del tejido – Que será representado por un punto "**.**"

Cada celda de la grilla va a estar relacionada con las celdas vecinas, cuya cantidad dependerá de la ubicación de la celda.

- Las celdas que ocupan las cuatro esquinas van a estar relacionadas con las 3 (tres) celdas vecinas
- Las celdas que ocupan los bordes de la grilla van a estar relacionadas con las 5 (cinco) celdas vecinas
- Las celdas del interior de la grilla van a estar relacionadas con las 8 (ocho) celdas vecinas.

Por ejemplo: Una grilla de 20 por 20 podría aparecer como sigue:

```
0 . . . . . . . . . . 0 . . . 1 . .
. . . . . 0 . . . . . . 1 . . . . .
. . 0 . . . . . . . . . . . . . . .
. . . . . . . . . 0 . . . . . . . 1
0 . . . . . . 0 0 1 1 0 . . . . . . 0
. . . . . . . . 1 2 0 . . . . . 1 .
. 0 . . . . . . . . 0 . . . . . . . .
1 . . 0 . . . . . . . . . . . . 1 . .
. . 0 . . . 1 . . 0 . 0 . 1 . . . 0 .
. . . . . . . . . . . . . . . . . . 0
0 . . . . . . . . . . . . . . . . . .
. . 0 . . . . . 0 . . . 0 . . . . . .
. . . . . . . . . . . . . 0 . . . . .
. . . . . . . . . . . 0 . . . . . . .
. . . 0 . . . . . . . . . . . . . 1 .
. . . . . . . . 1 . . . . . . . . . .
. . . . . . . . . . . 0 . . . . . . 0
. . 1 . . . . . . . . . . . 1 . . . 0
0 . 0 . 0 . . . . . . 1 . . . . 0 . .
. 0 . . . . . . . . 1 . . 0 . . . . . .
```

La simulación funciona calculando el nuevo estado de la grilla, a partir del estado de la grilla en el día anterior.

El "***nuevo estado de una célula"*** es calculado a partir del estado del día anterior de las células que están en las celdas vecinas en el día anterior, según las siguientes reglas:

Una célula sana puede infectarse de una vecina con probabilidad ***"p – propagación".*** Si una célula tiene cuatro vecinos enfermos, tiene cuatro oportunidades de infectarse. En ese caso, la probabilidad de no enfermarse es: $\mathbf{(1 - p)^4 = p^4 - 4p^3 + 6p^2 - 4p + 1}$ y la probabilidad de enfermarse es: $\mathbf{1 - (1 - p)^4} = \mathbf{- p^4 + 4p^3 - 6p^{2\,+}\,4p}$, pero es más fácil a determinar el estado de la célula después de la actualización con un loop de for en vez de derivar y calcular estas probabilidades.
Ejemplo: Si una célula tiene cuatro células vecinas que están enfermas, implica simular cuatro veces consecutivas la posibilidad de que se enferme. En este caso, la célula pasaría a estar "enferma" si en uno de los cuatro casos la probabilidad resultante así lo indica, o podría permanecer "sana" si luego de aplicar cuatro veces la probabilidad de propagación, en ninguno de los casos da un valor que indique contagio.

1. Una célula que está sana y que no tiene ningún vecino enfermo, permanecerá sana a lo largo del tiempo, ya que no hay nadie que la contagie.

2. Una célula vacía puede ganar un residente de un vecino. Tenemos un parámetro f – la fecundidad. Si una célula tiene un vecino, va a llenarse (un nacimiento) con probabilidad f. Una célula vacía con tres vecinos ocupados tiene tres oportunidades para ser llenado por un nacimiento. Un recién nacido nunca está enfermo, pero puede contagiarse el día siguiente.

3. Una célula enferma va a ser enferma y contagiosa por *d* días. Al final de ese periodo, tiene una probabilidad *m* – la *mortalidad* – de morir y la celda que ocupa pasará a estar vacía. Si no, su ocupante está sano y otra vez queda expuesto a un contagio.

Todos estos cálculos deben ser hechos de forma simultánea, lo que quiere decir que no se deben mezclar los datos ya actualizados con los datos que todavía no han sido actualizados.

Para generar la grilla de comienzo, se utilizarán las probabilidades; que se proporcionan como parámetros; a partir de los cuales se calculará el estado inicial de cada una de lasceldas de la grilla, el cual podrá ser:

- Celda Vacía – Que pasará a tener un punto " . "

- Celda ocupada por una célula sana que pasará a tener un cero " **0** "

- Celda ocupada por una célula enferma desde hace 1 (un) día, que pasará a tener un " **1** ".

La lista de los parámetros es la siguiente:

1. ***N***, el ***tamaño de la grilla***

2. ***"v"- Probabilidad de que una celda esté vacía en el estado inicial***

3. "***s" - Probabilidad de que una celda esté ocupada y sana en el estado inicial.***

4. "***e***" - ***Probabilidad que una celda esté ocupada y enferma en el estado inicial.***

5. "***f*" – *Probabilidad que una celda vacía pase a estar ocupada por un nacimiento de una célula, engendrada por una célula vecina sana o enferma.***

6. "***p***" – ***Probabilidad de que una célula sana se contagie***

7. "***m***" - ***Probabilidad de que muera una célula que estaba enferma.*** En este caso, la celda pasa a estar vacía.

8. "***d***" - ***número de días de la enfermedad***

9. "***t***" - ***número de días de la simulación***

Se deberá imprimir la nueva grilla de cada día que se obtiene a partir de la simulación.

En la impresión de cada día se deberá indicar el "***Nro. del día***" que corresponde, de la siguiente forma:

```
Starting grid:
. . 0 . 0 . 1 . . .
. 0 . . . . . . 0 0
. . 0 . . . . . . 0
. . . 0 . . . . 1 0
0 . . . . . . . 0 0
1 0 . . . . . . . 0
. 0 . . 0 . 0 1 . .
. . . . . . . . . .
. . . 0 . . 0 . . .
1 . . . . . 0 . 0 .

Grid at the end of day 1 of the simulation:
. . 0 . 0 . 2 0 . .
. 0 . . . . . . 0 0
. . 0 . . . . . . 0
. . . 0 . . . . 2 1
0 . . . . . . . 0 0
2 1 . . . . . . . 0
. 0 . . 0 . 0 2 . .
. . . . . . . 0 . .
. . . 0 . . 0 . . .
2 . . . . . 0 0 0 .
```

```
Grid at the end of day 2 of the simulation:
. . 0 . 0 0 3 1 . .
. 0 0 . . . . 0 0 0
. . 0 . . . . . . 1
. . . 0 . . . . . 3 2
0 . . . . . . . . 0 0
3 2 . . . . . . 0 . 0
0 0 . . 0 . 0 3 . .
. . . . . . . 0 0 . .
. . . 0 . 0 0 . . .
3 . . . . . 0 0 0 .

Grid at the end of day 3 of the simulation:
. . 0 . 0 0 4 2 0 .
. 0 0 . . . . 0 0 1
. . 0 . . . . . . 2
. . . 0 . . . . . 4 3
0 . . . 0 . . . 0 0
4 3 . . . . . 0 . 0
0 1 . . 0 . 0 4 0 .
. 0 . . . . . 0 1 . .
. . . 0 . 0 0 . . .
4 . . . . . 0 0 0 .
Grid at the end of day 4 of the simulation:
. . 0 . 0 0 0 3 0 .
. 0 0 . . . . 1 0 2
. . 0 . . . . . . 3
. . . 0 0 . . . . 4
0 . . . 0 . . . 0 1
. 4 . . . 0 . 0 . 0
0 2 . . 0 . 0 0 0 .
. 1 . . . . 0 1 . .
0 . . 0 . 0 0 . . .
0 . . . . . 0 0 0 .
Grid at the end of day 5 of the simulation:
. . 0 . 0 0 0 4 0 0
. 0 0 . . . . 2 0 1
. . 0 . . . . . . 4
. . . 0 0 0 . . . .
0 . . . 0 . . . 0 0
. . . . . 0 . 0 . 0
0 3 . . 0 . 1 0 0 .
. 2 . . . . 0 2 . .
0 . . 0 . 0 0 . . .
0 . . . . . 0 0 0 .
Grid at the end of day 6 of the simulation:
. . 0 . 0 0 1 . 0 0
. 0 0 . . . . 3 0 2
. . 0 . . . . . . 0
. . . 0 0 0 . . . .
0 . . . 0 . . . 0 0
. . . . . 1 . 1 . 0
0 4 . . 0 . 2 0 0 .
0 3 0 . . . 0 3 . .
0 . . 0 . 0 0 . . .
0 . . . . . 0 0 0 .
```

Ayudas:

- ✓ Una grilla puede ser representada mediante un diccionario, en el cual las claves son tuplas que representan las coordenadas “x” e “y” de cada una de las celdas de la grilla.
- ✓ Después de construir la nueva grilla, se puede copiar a la grilla actual con una asignación (grid = newGrid).
- ✓ El primer trabajo del obligatorio es el de implementar una función para imprimir la grilla. Esta función resultará de mucha utilidad para depurar el código.
- ✓ Implementar una función que calcule el nuevo estado de una célula de la grilla.
- ✓ No utilizar variables globales
- ✓ Comenzar con el diseño, a partir de las preguntas siguientes:
  - ¿Cuál son las reglas para actualizar una célula?
  - ¿Cuáles son las funciones **necesarias**?
- ✓ Probar las funciones individualmente.
- ✓ No todos los parámetros dan resultados interesantes.
  - Una enfermedad rápida y peligrosa no puede sostenerse porque mata a la población.
  - Una enfermedad con poca propagación tampoco puede sostenerse.
- ✓ Agregar un “***docstring”*** que explique el funcionamiento de cada función y clase que se defina.

Puntaje:

- Funcionamiento básico vale 90%.
- Tener un contador de células ocupadas y células enfermas vale 5% de puntuación actual.
- Parar la simulación si no hay población o si no hay enfermos vale 5% de la puntuación actual.
- Una versión GUI (con tkinter) da 50% de puntuación adicional.

Entrega:

1) El obligatorio se entregará siempre por medio dela web asignatura, **salvo que el profesor de cada grupo solicite expresamente otro mecanismo de entrega**.

2) En forma opcional, los que deseen pueden entregar un documento (carpeta) en formato Word o PDF, donde se explique la solución implementada. Sugerimos que dicho documento tenga la estructura:

- Introducción
- Objetivo del trabajo
- Detalle de la solución
- Ejecución del programa
- Comparación con resultados obtenidos de otra fuente (este punto va si aplica, según el obligatorio propuesto)
- Conclusiones finales

Los que presenten dicho documento tendrán una bonificación de un 10% adicional en la nota final del trabajo.

3) Deben subir a la web un único archivo por grupo, en formato zip o rar.

El archivo deberá nombrarse con los apellidos de los integrantes del grupo, separados por guiones, por ejemplo:

rodriguez-gonzalez.zip

El zip o rar deberá ser sin contraseña y deberá incluir el archivo .py, el documento (carpeta) si lo hubiera y los archivos generados en la ejecución de los distintos puntos del obligatorio (si corresponde).

Deben asegurarse que el docente pueda ejecutar el programa en las mismas condiciones que Uds. lo hicieron, por lo que se sugiere no usar path estáticos en los archivos.

Tengan a bien pasarle antivirus al archivo zip/rar antes de subirlo al web asignatura

Buena suerte!