



DESARROLLO DE ALGORITMOS

Evaluación

UCO
ONLINE



El juego de la vida

Objetivos

Desarrollar un programa en Python que haga uso de los contenidos vistos hasta el momento.

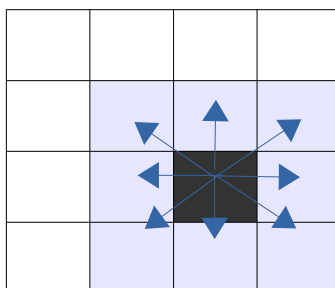
Temporización

180 minutos

Enunciado

Descripción

Consideremos una población de K insectos en una matriz de dimensiones $(M \times N)$, de modo que en cada celda de la matriz hay, como máximo, un insecto. Por lo tanto, cada insecto tiene, como máximo, 8 insectos vecinos.



La población está en desarrollo continuamente debido a los nacimientos y muertes que se producen. Las reglas de evolución que se observan son las siguientes:

1. Aquellos insectos que tienen 0, 1, 4, 5, 6, 7 u 8 vecinos mueren.
2. Los insectos que tienen 2 o 3 vecinos sobreviven.
3. En las celdas vacías en cuya vecindad haya 2 o 3 insectos nace un nuevo insecto.

Los insectos que nacen o mueren no afectan a las reglas hasta que se ha completado un ciclo evolutivo, entendiendo por éste un ciclo en el que se ha decidido la supervivencia o muerte de los insectos (vivos al comenzar el ciclo) de acuerdo a las reglas mencionadas. Esto quiere decir que al ir calculando los nuevos estados de las celdas de la matriz las

modificaciones se deben hacer sobre una matriz auxiliar que, finalmente, se copiará en la matriz original.

Objetivo

Escribir un programa que haga uso de la función *Evolución* para la simulación de una población de insectos. La función *Evolucion* simulará la evolución de la población y :

- Recibirá como entrada los enteros positivos K (nº de insectos en la población inicial), N (nº de filas de la matriz) y M (nº de columnas de la matriz), una *lista* con las coordenadas x e y de las celdas en las que se encuentran los K insectos iniciales de la población y un entero positivo L que indica cuántos ciclos se van a simular.
- Rellenará la población inicial a partir de la lista de coordenadas.
- Mostrará por pantalla el estado inicial de la población, marcando con un asterisco (*) las celdas ocupadas por insectos
- Simulará la evolución de la población durante los L ciclos evolutivos y representará en la pantalla el estado de la población en cada ciclo mostrándolo como en el siguiente ejemplo:

Ejemplo

*Matriz de 10*10, 3 insectos colocados inicialmente en las posiciones (0,0), (0,1) y (1,0) y 3 ciclos de evolución*

Configuración inicial:

```

| ** |
| *  |
|    |
|    |
|    |
|    |
|    |
|    |
|    |
|    |

```

Primera iteración:

```

| ** | * |
| ** | * |
|    |   |
|    |   |
|    |   |
|    |   |
|    |   |
|    |   |
| ** |   |

```

Segunda iteración:

```

|  *      *  |
|  **     ** |
| **      *  |
|          |
|          |
|          |
| **      *  |
| **      *  |
|          |

```

Tercera iteración:

```

|  *      ** |
|  **     ** |
|  **     ** |
| **      *  |
| **      *  |
|          |
|          |
| **      *  |
| ***     *  |
| **      ** |

```

Funciones auxiliares

El programa deberá implementar al menos las siguientes funciones

rellena_poblacion_inicial. Función que recibe la lista con las posiciones iniciales de los insectos e inicializa la matriz de insectos vivos de acuerdo a dichas posiciones

visualiza. Función que recibe la matriz con los insectos y los datos necesarios para pintarla por pantalla utilizando asteriscos, tal y como muestra el ejemplo anterior.

cuenta_vecinos. Función que recibe la matriz de insectos, las coordenadas de una posición dentro de la matriz y los datos necesarios para devolver el nº de insectos vecinos a esa posición. Hay que considerar la matriz circular, es decir, si tenemos una matriz de 4x4 y estamos inspeccionando la casilla [3][3] (esquina inferior derecha), los vecinos serán las casillas [2][2], [2][3], [2][0], [3][2], [3][0], [0][2], [0][3], y [0][0].

(0,0)		(0,2)	(0,3)
(2,0)		(2,2)	(2,3)
(3,0)		(3,2)	(3,3)

proxima_generacion. Función que recibe la matriz de insectos y los datos necesarios para devolver, la nueva población de insectos después de aplicar las reglas del juego de la vida (generación).