

Incendio Forestal

Pensamiento Computacional

Facultad de Ciencias Exactas y Naturales, UBA

Segundo Cuatrimestre 2024

ENCUESTA CLASE ANTERIOR Y REPASO

Encuesta de cierre de clase anterior

Sólo si asistieron la última clase y no habían completado la encuesta de cierre, les pedimos que se tomen unos minutos para completarla antes de comenzar la nueva clase.

Les pedimos total sinceridad para compartir sus sensaciones de cómo se llevaron con la clase. **No hay respuestas correctas o incorrectas.**



bit.ly/pc-encuesta-clase

Repaso de algunos conceptos de la clase anterior

- Recorrido de una lista usando `while`.
- Recorrido de una lista usando `for`.
- Funciones que devuelven valores booleanos.

INCENDIO FORESTAL

Incendio Forestal



 **INCENDIO DE COPAS**

Sistemas complejos

- Sistemas en los cuales el comportamiento colectivo es difícil de anticipar a partir del comportamiento individual de sus elementos.
- Las reglas a nivel microscópico pueden ser muy simples pero generar propiedades emergentes a nivel macroscópico sorprendentes.

La computadora nos permite programar, simular, explorar y detectar estas cosas.





La verdad sobre China: ¿El embotellamiento más largo del mundo duró 12 días?

Daniel Canal | 13-01-2015 - 05:40:58

Mito: En China el tráfico vehicular es tal que la peor congestión registrada en la historia ocurrió en Beijing y duró 12 días.

Realidad: Cuando el conductor de un Peugeot 404 pasa de segunda marcha a tercera del cuento "Autopista al sur", de Julio Cortázar, se produce un embotellamiento que dura lo suficiente para...

El 14 de agosto de 2010 empezó el embotellamiento más demorado en la historia, en la autopista 1 que conecta a Beijing, la capital china, con el Tibet. Este atasco se demoró 12 días en descongestionarse y se extendió a lo largo de 100 kilómetros. Lo particular es que el embotellamiento ocurrió sin ninguna razón aparente, solo porque había muchos autos en la vía, contrario al de París o Chicago como el de Chicago en 2011, que fueron consecuencia del mal clima.

Veredicto: Sí, el embotellamiento más demorado del mundo ocurrió en China en una carretera al sur de Beijing. Los chinos debieron vivir casi dos semanas en sus autos, y como el protagonista de Cortázar, cuando se abren los autos y pasaron de segunda a tercera marcha, dejaron su vida...







¿Por qué incendios de bosques?

Fue uno de los primeros modelos (~ 1992), con el de avalanchas en pilas de arena.

Buscaba modelar la dinámica de un bosque, que consiste en el crecimiento de árboles y la aparición de incendios que los eliminan. Entender por qué los bosques tienen los tamaños que tienen, y se alternan sectores con árboles y sectores sin árboles.

Hay mil variantes, pero ninguna explica ningún bosque conocido. ¿Por qué no nos olvidamos del modelo, entonces?

Porque se aplica en otros problemas, como epidemias o la difusión de noticias/rumores...

El modelo (primera aproximación)

Modelaremos ciclos de un año donde a cada posición del bosque le suceden cosas.

Bosque es un segmento lineal, dividido en n celdas, y cada una puede:

- tener un árbol
- no tener nada

Durante cada año se suceden una serie de **etapas**:

- Brotes
- Caída de rayos
- Propagación de incendios
- Limpieza

Etapas

- 1 Brotes: en cada celda vacía, puede brotar un nuevo árbol (o no).
- 2 Caída de rayos: caen rayos en algunas posiciones, incendiando a sus árboles.
- 3 Propagación de incendios: Cuando un árbol se incendia, propaga el fuego a los árboles de celdas vecinas, y estos, a su vez, a su vecinos... y así, hasta que todo árbol incendiado incendió a todos sus vecinos y el incendio no puede propagarse más.
- 4 Limpieza: los árboles incendiados degradan y dejan la posición vacía, lista para que vuelva a comenzar el ciclo.

Experimentando la dinámica


















Pre-requisitos:

- Una computadora donde poder correr `random.randint(a,b)`, donde a es el número más bajo que puede salir en el dado y b el más alto. Ahora vamos a imaginar que tenemos un dado con 100 caras. Es decir que $a=1$ y $b=100$
- Una tarjeta blanca, una verde y una roja.
- Determinar quiénes son tus vecinos.

A simular


















- Todos empiezan siendo celdas vacías.
- Todos los que saquen entre **1** y **80** (inclusive) se transforman en árboles.
- Todos los que saquen entre **1** y **30** (inclusive) se prenden fuego.
- A propagar.
- Limpiar.
- Otra vez.

Etapas de la dinámica de un bosque - Año 0

Inicio						
Brotes						
Rayos						
Propagación						
						
Limpieza						



























Al final del primer año: 1 árbol.

Etapas de la dinámica de un bosque - Año 1

Inicio						
Brotes						
Rayos						
Propagación						
						
Limpieza						































Al final del segundo año: 0 árboles.

Etapas de la dinámica de un bosque - Año 2

Inicio						
Brotes						
Rayos						
Propagación						
						
						
Limpieza						

Al final del tercer año: 4 árboles.

Etapas de la dinámica de un bosque - Año 3

Inicio							
Brotes							
Rayos							
Propagación							
							
							
Limpieza							

Al final del cuarto año: 1 árbol.

Modelo de incendios de bosques

Queremos tomar los ingredientes básicos del problema real y construir un modelo.

- **Bosque:** tenemos n celdas, y en cada una puede crecer un solo árbol.
- **Brotes:** brota un árbol en cada celda vacía, con probabilidad p (la calidad del terreno).
- **Caída de rayos:** cae un rayo en cada celda con probabilidad f . Si la celda tiene un árbol, se prende fuego.
- **Propagación de incendios:** si un árbol está prendido fuego y tiene al lado un árbol sano, éste también se prende fuego. Se termina cuando no queda ningún árbol sano con un vecino prendido fuego.
- **Limpieza:** los árboles quemados se retiran dejando la celda vacía.

Representación

¿Cómo representar esto en la máquina?

- **Bosque:** tomamos una arreglo unidimensional de longitud n , indexadas del 0 al $n - 1$.
- **árboles:** representamos las posiciones como:
 - 0 si está vacía,
 - 1 si hay un árbol vivo,
 - 1 si hay un árbol prendido fuego.

Problemas interesantes a resolver hoy:

- 1 ¿Cuál es el valor de p que maximiza la cantidad de árboles que sobreviven cada año?
- 2 Si cada posición pudiera tener y modificar su propio valor de p , ¿qué pasaría?
- 3 ¿Y si modelamos un sistema dónde la propagación no sólo se da entre vecinos?

¡A trabajar!

Sugerencias: lean cada consigna ENTERA antes de empezar a escribir. Lean las ideas y prueben las implementaciones. Dentro de una función se puede invocar a otra definida previamente. Recuerden importar los módulos a utilizar.

Lista de funciones:

- **def generar_bosque(n):** genera un bosque vacío del tamaño indicado.
- **def cuantos(bosque, tipo_celda):** recibe cualquier tipo de bosque, cuenta cuantas veces aparece `tipo_celda` dentro del bosque (hay que recorrer el bosque, ciclos). Devuelve la cantidad de apariciones que contó.
- **def suceso_aleatorio(p):** función auxiliar que debe devolver True con probabilidad p , o False con probabilidad $100-p$.

CORTE!

SEGUIMOS TRABAJANDO

¡A trabajar!

- **def brotes(bosque, p):** recibe un bosque **CUALQUIERA**, recorre cada posición del bosque (ciclos) y brota o no según la posición se encuentre vacía, con probabilidad de brote p . ¡Devuelve el bosque modificado!
- **def rayos(bosque, f):** análoga a la función “brotes”.

¡A trabajar!

- **def propagacion(bosque)**: recibe un bosque cualquiera. Cada árbol quemado, quema a sus vecinos ya sea hacia la derecha o izquierda. Devuelve el bosque quemado.
- **def limpieza(bosque)**: elimina todos los árboles quemados del bosque, dejando celdas vacías en su lugar.
- **def dinamica(n,a,p,f)**: invoca a todas las funciones anteriores. **El orden importa.** Utilicen a los parámetros n,a,p,f para invocar a las funciones previas.
 - generar bosque vacío
 - repetir **a** veces la siguiente secuencia de etapas:
 - brotes
 - rayos
 - propagación
 - limpieza
 - **CONTAR CUANTOS ÁRBOLES QUEDARON VIVOS**
 - Calcular y devolver el promedio de árboles vivos al final de cada año.

Ideas para la tarea (ejercicios 8 y 9)

Simulaciones: armamos una grilla de 100 lugares y tomemos un p arbitrario. Entonces, en el bosque...

- brota un árbol con probabilidad p en cada lugar vacío, cae un rayo con probabilidad 2 %, se propaga, limpiamos los quemados, y contamos cuántos quedan. En el bosque que queda...
- brota un árbol con probabilidad p en cada lugar vacío, cae un rayo con probabilidad 2 %, se propaga, limpiamos los quemados, y contamos cuántos quedan. En el bosque que queda...
- brota un árbol con probabilidad p en cada lugar vacío, cae un rayo con probabilidad 2 %, se propaga, limpiamos los quemados, y contamos cuántos quedan. En el bosque que queda...

Lo hacemos 1000 veces, y calculamos el promedio.

Ideas para la tarea (ejercicios 8 y 9)

Y esto lo hacemos explorando valores de p entre 0 % y 100 %:

```
probs = []  
promedios = []  
for p in range(0, 101):  
    ...COMPLETAR...  
    probs.append(p)  
    promedios.append(...COMPLETAR...)
```

Ideas para la tarea (ejercicios 8 y 9)

Ahora buscamos cuándo se alcanza el máximo ¿Cómo?

¡Graficando!

Para cada probabilidad marcamos con un punto en el plano la cantidad de árboles sobrevivientes.

¿Puede `Python` ayudarnos con esta tarea?

¡Claro! ¡Con `Matplotlib`!

Matplotlib.Pyplot

- Al igual que `random` o `numpy`, es un módulo que podemos importar para hacer uso de sus funciones.
- Matplotlib nos permite elegir una lista de coordenadas sobre el eje x , otra lista sobre el eje y , y **graficar**.
- Para importarlo, hay que usar

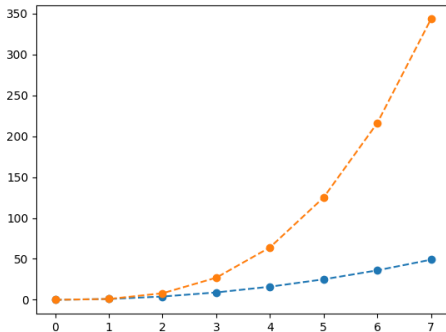
```
import matplotlib.pyplot as plt
```

Comandos para graficar (1/2)

```
import matplotlib.pyplot as plt

x = [0, 1, 2, 3, 4, 5, 6, 7]
y1 = [0, 1, 4, 9, 16, 25, 36, 49]
y2 = [0, 1, 8, 27, 64, 125, 216, 343]
plt.plot(x, y1, "--o")
plt.plot(x, y2, "--o")
plt.show()
```

Este código hace el siguiente dibujo:



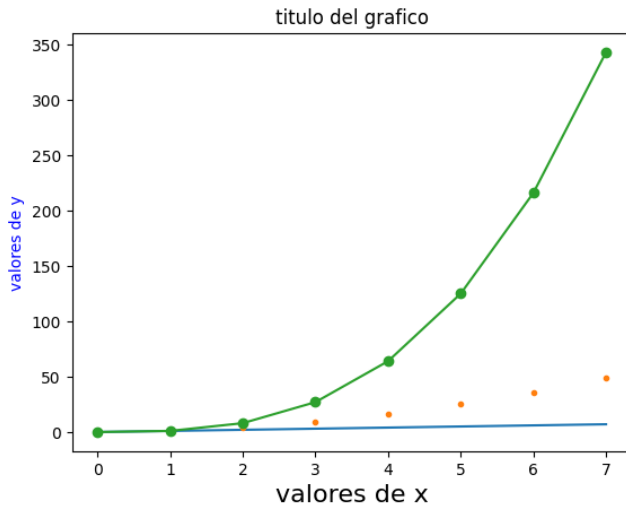
Comandos para graficar (2/2)

Adicionalmente, podemos definir el título, el nombre de los ejes, el tipo de gráfico, etc. Por ejemplo, en nuestro caso:

```
plt.title("titulo del grafico")
plt.xlabel("valores de x", fontsize = 16)
plt.ylabel("valores de y", color = "blue")
plt.plot(x, x, "-")
plt.plot(x, y1, ".")
plt.plot(x, y2, "-o")
plt.show()
```

¡No se olviden de importar `random` y `numpy` además de `pyplot`!

Gráficos



Generando datos para graficar

El mismo ejemplo, pero usando un ciclo para generar las listas elemento a elemento (les puede ser muy útil ;)

```
import matplotlib.pyplot as plt
```

```
x = []  
y1 = []  
y2 = []  
numero = 0  
numero_final = 7  
while numero <= numero_final:  
    x.append(numero)  
    y1.append(numero ** 2)  
    y2.append(numero ** 3)  
    numero = numero + 1
```

```
plt.plot(x, y1, "--o")  
plt.plot(x, y2, "--o")  
plt.show()
```

CIERRE DE CLASE

Antes de irse del aula

No se olviden de...

- 1 revisar que en su resolución tienen las funciones pedidas **con el nombre y parámetros solicitados**,
- 2 **guardar el archivo de su resolución** con un nombre con la extensión `.py` (por ej., `clase04.py`),
- 3 **subir la resolución al Campus**, en la solapa “TAREAS”, tarea “Incendio - Entrega de fin de clase”.
- 4 Además, **deberán completar la encuesta** del cierre de la clase:



bit.ly/pc-encuesta-clase