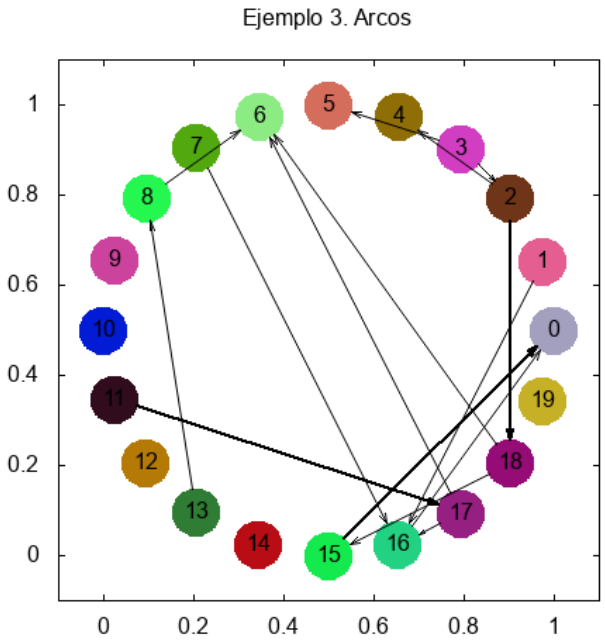
jbenavidesv87 Clarificada la explicación del ciclo en el reporte del ejemplo 3. Latest commit da99f0f Mar 1, 2018

..		
grafo.gnu	Reporte del Ejemplo 3, terminado.	Mar 1, 2018
grafo.png	Reporte del Ejemplo 3, terminado.	Mar 1, 2018
main.py	Reporte del Ejemplo 3, terminado.	Mar 1, 2018
readme.md	Clarificada la explicación del ciclo en el reporte del ejemplo 3.	Mar 1, 2018

readme.md

Ejemplo 3. Arcos

En este programa se crea un grafo dirigido con veinte nodos coloreados al azar dispuestos en forma circular en torno a un centro en (0.5, 0.5), identificados cada uno por números enteros del 0 al 19, ambos incluidos. Cada uno de estos veinte nodos se conecta con una probabilidad de 0.3 con otros tres nodos elegidos al azar. A cada vecindad establecida de esta manera, se le asigna un peso al azar en el rango [0, 2] que se refleja en el groso de la línea que representa cada arco. Uno de los grafos resultantes es el siguiente:



El código para hacerlo requiere el uso de las funciones `random`, `randint` y `sample` de la librería `random`, y de las funciones `sin` y `cos` de la librería `math`.

```
from random import random, randint, sample
from math import cos, sin
```

La función `sample(a, b)` regresa una lista de *b* elementos elegidos al azar de la población *a*. Las funciones `sin(c)` y `cos(c)` devuelven el seno y coseno de *c* en radianes.

Se establecen:

- $N = 20$ nodos
- Un radio $r = 0.5$ para la circunferencia que formarán los nodos entre sí
- Un perímetro $P = 2 \times 3.14 \times r$ de la circunferencia que formarán los nodos entre sí
- Un radio $r_n = P / N / 3$ para cada nodo. En este caso, se divide P / N para conocer la medida del arco correspondiente a cada nodo y se divide ese valor entre 3 para que las circunferencias de los nodos no coincidan con los centros de los nodos adyacentes. (Recomiendo eliminar esa división entre tres para ver el cambio mencionado)
- Una fracción angular $\theta = 2 \times 3.14$ rad que ocupará cada nodo respecto a la circunferencia que se formará
- Una probabilidad $p_v = 0.3$ de establecerse vecindades

Estos parámetros corresponden respectivamente con el siguiente código en `python`:

```
N = 20
r = 0.5
P = 2 * 3.14 * r
rNodo = P / N / 3
theta = 2 * 3.14 / N
pVecino = 0.3
```

Ahora se instancia un grafo *G* y se especifica que sea dirigido:

```
G = Grafo()
G.dirigido = True
```

Inmediatamente después, se hace un ciclo de $i \in [0, 1, 2, ..., N - 1]$, donde se crean N nodos, asignándosele a cada nodo n_i un identificador i, un radio igual a r_n, un color RGB al azar y una posición (x_i, y_i) a lo largo de la circunferencia que formarán que dependerá de las funciones

$x_i = 0.5 + r \times \cos(\theta i)$

$y_i = 0.5 + r \times \text{sen}(\theta i)$

Y se agregará cada nodo n_i al grafo G, todo lo cual corresponde al código:

```
for i in range(N):
    n = Nodo()
    n.id = i
    n.radio = rNodo
    n.Color(
        randint(0, 255), # rojo (R)
        randint(0, 255), # verde (G)
        randint(0, 255),  # azul (B)
    )
    n.posicion = (
        0.5 + r * cos(theta * i), 0.5 + r * sin(theta * i)
    )
    G.AgregarNodo(n)
```

Finalmente, por cada nodo n_i dado $i \in [0, 1, ..., N - 1]$ y con una probabilidad de p_v se intenta establecer una vecindad con algún nodo al azar entre el resto de los nodos del grafo G. Esta probabilidad de realiza tres veces por cada nodo, eligiendo en cada ocasión un nodo al azar entre los restantes. Luego, por cada intento exitoso de establecer una vecindad, se elige un vecino v al azar y se conecta con el nodo n_i correspondiente con un peso elegido al azar en un rango [0; 2]. Lo anterior se realiza en el siguiente ciclo:

```
for n in G.nodos:
    for i in range(3):
        if random() < pVecino:
            candidatos = set(G.nodos) - set([n]) # Se quita el nodo n de todo los nodos
            v = sample(candidatos, 1) # Se elige un nodo al azar
            G.ConectarNodos(n, v[0], random() * 2) # Se establece la vecindad dirigida de n a v con peso
```

Por último, se dibuja el nodo con título **Ejemplo 3. Arcos**:

```
G.DibujarGrafo("Ejemplo 3. Arcos")
```