



PUCMM

Pontificia Universidad Católica
Madre y Maestra

Facultad de Ciencias de las Ingenierías

Departamento de Ingeniería de Sistemas y Computación

Tarea 4 (T4):

Coloración de Mapas

Por:

Ansël E. Corona Coste 2014-0031

Profesor:

J.R Felipe Nuñez

Asignatura:

Diseño y Análisis de Algoritmos

ISC-405-T-001

Santiago de los Caballeros

República Dominicana

Marco teórico:

En Teoría de grafos, la coloración de grafos es un caso especial de etiquetado de grafos; es una asignación de etiquetas llamadas colores a elementos del grafo. De manera simple, una coloración de los vértices de un grafo tal que ningún vértice adyacente comparta el mismo color es llamado vértice coloración. Similarmente, una arista coloración asigna colores a cada arista tal que aristas adyacentes no compartan el mismo color, y una coloración de caras de un grafo plano a la asignación de un color a cada cara o región tal que caras que compartan una frontera común tengan colores diferentes. El vértice coloración es el punto de inicio de la coloración, y los otros problemas de coloreo pueden ser transformados a una versión con vértices. Por ejemplo, una arista coloración de un grafo es justamente una vértice coloración del grafo línea respectivo, y una coloración de caras de un grafo plano es una vértice coloración del grafo dual.

Para este problema tendremos 4 colores (Rojo, Azul, Amarillo y Verde), los cuales utilizaremos para colorear un mapa de Europa, el que recibimos en un archivo que describe cual país es vecino de otro.

Código:

```
colores = ['Azul', 'Rojo', 'Amarillo', 'Verde']
vecinos = {}
países = []
paisVisto = {}

mapa = open('Mapa.txt', 'r')

for linea in mapa:
    line = []
    for x in linea.split():
        line.append(x)
    vecinos.update({line[0]: line[1:]})

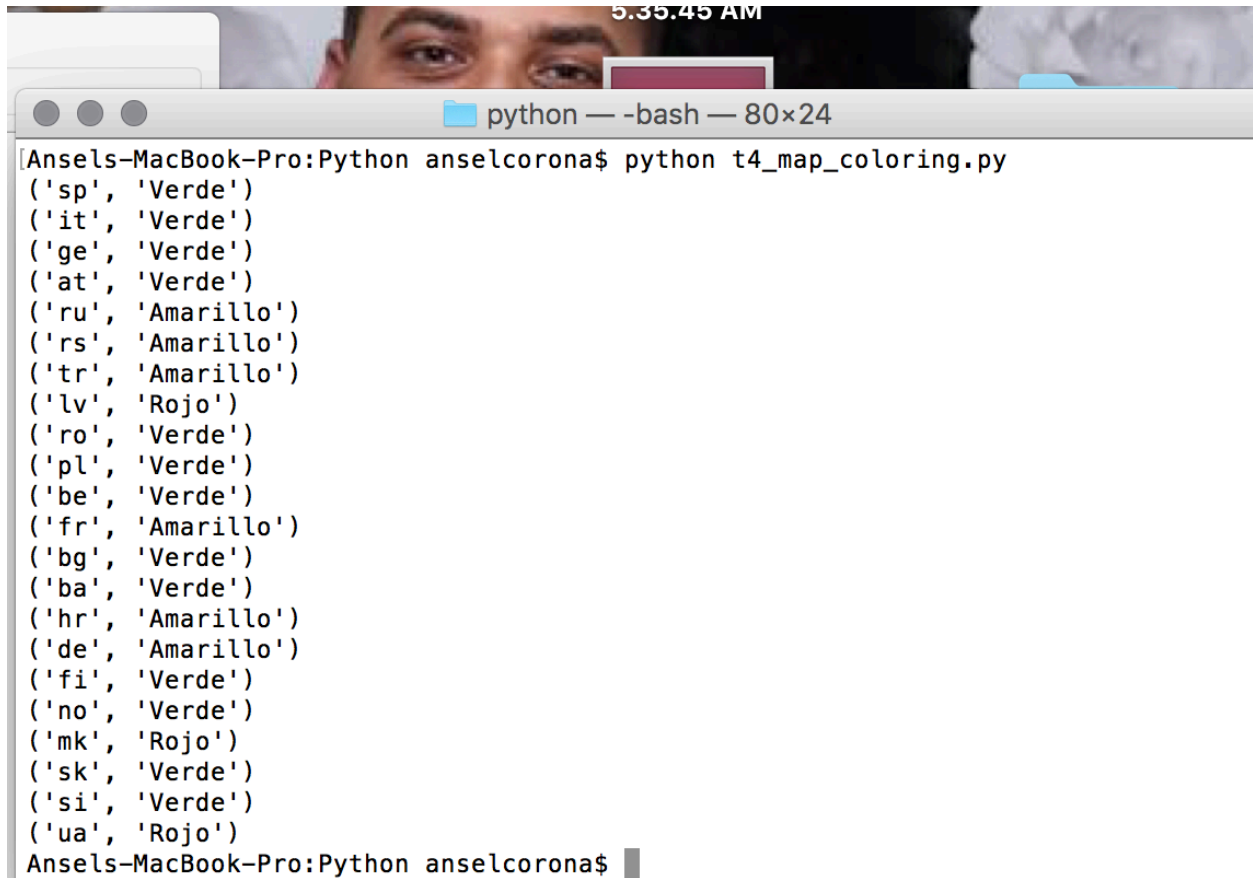
for pais in vecinos.keys():
    países.append(pais)

def esVecino(pais, color):
    for vecino in vecinos.get(pais):
        colorPaisActual = paisVisto.get(vecino)
        if colorPaisActual == color:
            return False
    return True

for pais in países:
    for color in colores:
        if esVecino(pais, color):
            paisVisto[pais] = color

for k in paisVisto:
    print(k, paisVisto[k])
```

Conclusión:



```
Ansels-MacBook-Pro:Python anselcorona$ python t4_map_coloring.py
('sp', 'Verde')
('it', 'Verde')
('ge', 'Verde')
('at', 'Verde')
('ru', 'Amarillo')
('rs', 'Amarillo')
('tr', 'Amarillo')
('lv', 'Rojo')
('ro', 'Verde')
('pl', 'Verde')
('be', 'Verde')
('fr', 'Amarillo')
('bg', 'Verde')
('ba', 'Verde')
('hr', 'Amarillo')
('de', 'Amarillo')
('fi', 'Verde')
('no', 'Verde')
('mk', 'Rojo')
('sk', 'Verde')
('si', 'Verde')
('ua', 'Rojo')
Ansels-MacBook-Pro:Python anselcorona$
```

La respuesta es mostrada.

En fin la solución es bien sencilla, lo que hay que hacer es iterar sobre los vecinos e ir asignando colores a los países verificando si tienen un vecino con el mismo color.

Python de nuevo 😊

T4✓✓