

Pontificia Universidad Católica Madre y Maestra
Facultad de Ciencias de la Ingeniería

Tarea #4:

Coloreo de Mapas

Presentado Por:

Angel González (2013-0157)

Materia:

Diseño y Análisis de Algoritmos

Profesor:

Juan Núñez.

Marco Teorico

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.

La convención de usar colores se origina de la coloración de países de un mapa, donde cada cara es literalmente coloreada. Esto fue generalizado a la coloración de caras de grafos inmersos en el plano. En representaciones matemáticas y computacionales se utilizan típicamente enteros no negativos como colores. En general se puede usar un conjunto finito como conjunto de colores. La naturaleza del problema de coloración depende del número de colores pero no sobre cuales son.

Codigo

```
import time
colors = ['rosado','violeta', 'fusia', 'verde luminico']
neighbors = {}
countries = []
countriesSeen = {}

start_time = time.time()

mapa = open('Mapa.txt', 'r')
for linea in mapa:
    line = []
    for x in linea.split():
        line.append(x)
    neighbors.update( {line[0]: line[1:]} )
for pais in neighbors.keys():
    countries.append(pais)

def areNeighbors(pais, color):
    for neighbor in neighbors.get(pais):
        colorPaisActual = countriesSeen.get(neighbor)
        if colorPaisActual == color:
            return False
    return True

for pais in countries:
    for color in colors:
        if areNeighbors(pais, color):
            countriesSeen[pais] = color

for k in countriesSeen:
    print(k, countriesSeen[k])

print("--- %s seconds ---" % (time.time() - start_time))
```

Corrida

('sp', 'verde luminico')

('it', 'verde luminico')

('ge', 'verde luminico')

('at', 'verde luminico')

('ru', 'fusia')

('rs', 'fusia')

('tr', 'fusia')

('lv', 'violeta')

('ro', 'verde luminico')

('pl', 'verde luminico')

('be', 'verde luminico')

('fr', 'fusia')

('bg', 'verde luminico')

('ba', 'verde luminico')

('hr', 'fusia')

('de', 'fusia')

('fi', 'verde luminico')

('no', 'verde luminico')

('mk', 'violeta')

('sk', 'verde luminico')

('si', 'verde luminico')

('ua', 'violeta')

--- 0.00548100471497 seconds ---