Morgan G. Vásquez
Ingeniero mecatrónico 06/06/2019

La figura muestra un mapa con 4 distritos A, B, C y D. Se trata de pintar cada distrito con un color de forma que, dos regiones con un borde común (que no sea un punto) tengan distintos colores y queremos hacer esto usando un mínimo de colores.

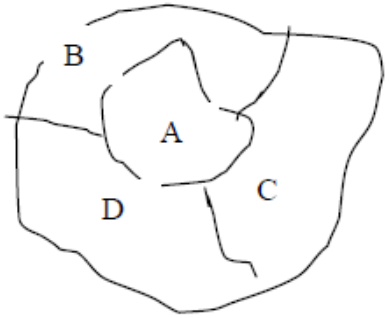

Figura 1. Mapa con 4 distritos.

1. Encuentra una representación en términos de vértices y aristas de un grafo a partir del mapa dado.

Cada área del mapa se representará por un vértice. Para representar las aristas, debemos observar el mapa y ver los vértices adyacentes a dicho vértice. Por ejemplo para A, podemos ver que B,C y D son adyacentes. Por lo tanto debemos unir los vértices B,C y D mediante líneas a A, lo cual representará las aristas y para etiquetarla usaremos la letra e. Haremos lo mismo para las otras aristas.

En la Figura 2 podemos ver los vértices A,B,C,D y las aristas: e1,e2,e3,e4,e5,e6

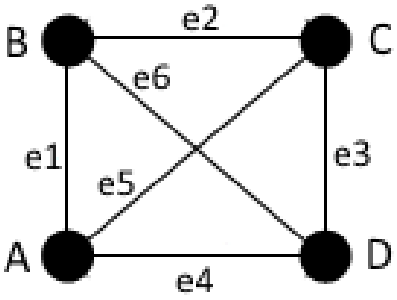

Figura 2. Representación del mapa en vértices y aristas.

Morgan G. Vásquez
Ingeniero mecatrónico 06/06/2019

2. Investiga un algoritmo que aplicado a grafos te permita ir coloreando los vértices de tal forma que no coincidan en color, con el color de los vértices que estén unidos a ellos a través de aristas.

Los siguientes algoritmos solucionan el problema del coloreado de grafos:

1. Algoritmo Secuencial o Voraz
2. Algoritmo de coloración Welsh- Powell
3. Algoritmo de coloración Matula-Marble-Isaccson

ALGORITMO SECUENCIAL O VORAZ

Este algoritmo sigue una estrategia voraz, es decir comienza la coloración de los vértices según orden de los éstos en la matriz de adyacencias del grafo. La coloración se realiza siguiendo los siguientes pasos.

1. Ingresamos el número de nodos y las aristas, se llenan los datos y creamos la matriz de adyacencias.

2. (Inicia Algoritmo Secuencial)Se tienen ordenados los vértices de 1 a n, y se selecciona el primero en la lista y se colorea o se etiqueta con el 1.

3. Se toma el siguiente vértice y se colorea o etiqueta con en el menor número admisible; es decir, se verifican adyacencias.

4. Se repite el paso (3) hasta que todos los vértices hayan sido coloreados.

5. Se imprime la solución.

Ejemplo:

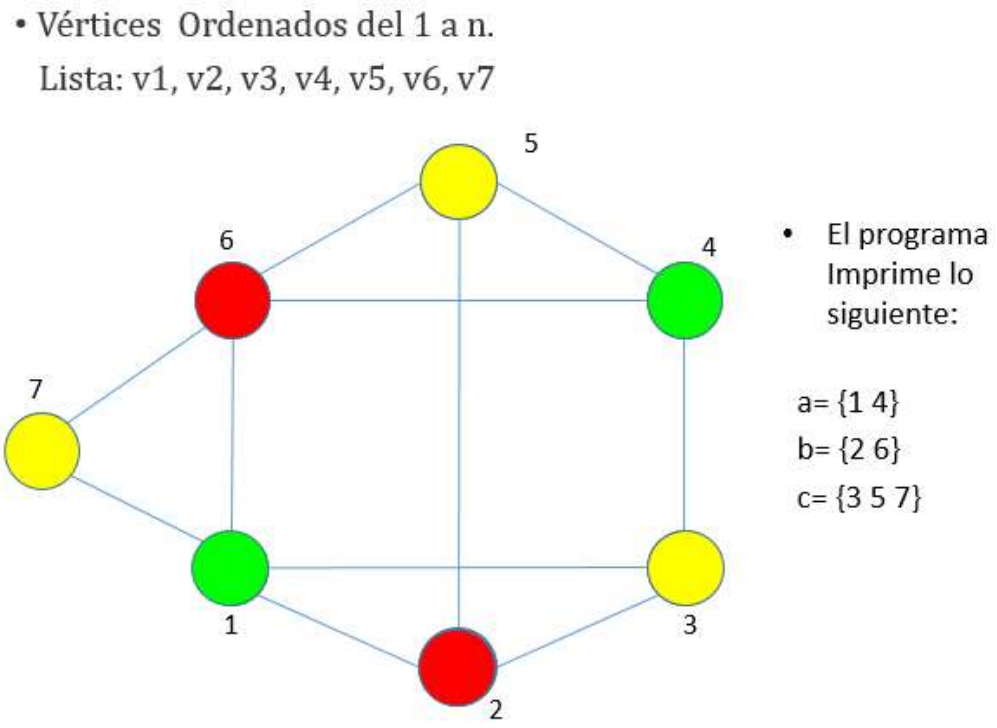

Figura 3. Ejemplo de algoritmo voraz

3. Como resultado presenta un documento en formato Word que ofrezca la explicación del algoritmo de coloración que hayas utilizado, en conjunto con la corrida a mano de la coloración del grafo, la cual representa al mapa dado en la actividad.

Usaremos el algoritmo voraz para colorear el grafo. El teorema para colorear un mapa dice que con cuatro colores es suficiente. Elegiremos los colores amarillo, rojo, verde y azul.

Para empezar definimos los vértices y aristas de la siguiente forma:

v={A,B,C,D}

ar={(A,B),(B,C),(C,D),(D,A),(A,C),(B,D)}

ar representa las aristas, donde cada arista está conectada a los vértices, por ejemplo la arista e1 contiene en sus extremos los vértices A y B.

El primer vértice (A) puede ser coloreado con el primer color que es el amarillo.

El segundo vértice (B) ya no puede ser coloreado de amarillo porque es adyacente a (A) tiene que ser coloreado con el siguiente color de la lista, que será el rojo para que sea una coloración propia.

El tercer vértice (C) es adyacente con los vértices (A) y (B), no se puede colorear ni de amarillo ni de rojo, se coloreará con el tercer color que es el verde.

Por último el vértice (D) es adyacente a los vértices (A),(B) y (C), por lo que se coloreará con el último color, que es el azul.

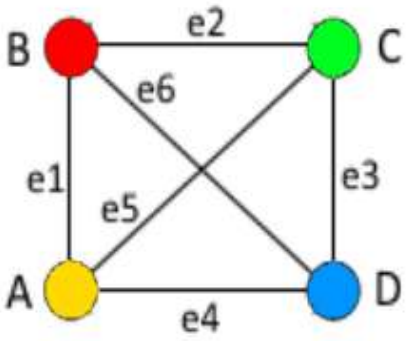

Figura 4. Grafo coloreado

El número cromático hace referencia a la cantidad mínima de colores necesarios para colorear el grafo, para el mapa necesitamos 4 colores, por lo que X(G)=4, el número cromático del grafo es igual a 4.

En la Tabla 1 podemos ver los grados de los vértices, esto se refiere a la cantidad de vértices adyacentes a dicho vértice.

Grados	3	3	3	3
Vértices	A	B	C	D
Color				

Tabla 1. Lista de vértices y grados de los vértices

Morgan G. Vásquez
Ingeniero mecatrónico 06/06/2019

VISUALIZACIÓN DEL GRAFO EN PYTHON

Para poder dibujar el grafo en Python necesitaremos la librería Networkx y matplotlib. El grafo se dibujará siguiendo los nombres de nodos y aristas del ejercicio. A continuación se presenta el código:

```
# -*- coding: utf8 -*-
# Python 3.7
# @author 3ngineer2k8

import networkx as nx # librería para usar grafos
import matplotlib.pyplot as plt # librería para dibujar
G = nx.Graph() # se crea un grafo vacío sin nodos y aristas
H = nx.path_graph(4) # 4 nodos
G.add_nodes_from(H) # lista de nodos (A,B,C,D) donde A=0 B=1 C=2 D=3
G.add_edges_from([(0,1),(1,2),(2,3),(3,0),(0,2),(1,3)]) # lista de aristas
#               ([(A,B),(B,C),(C,D),(D,A),(A,C),(B,D)])
#               ([(e1 ),(e2 ),(e3 ),(e4 ),(e5 ),(e6 )])
# ver cantidad de nodos y aristas
print('Cantidad de nodos:',G.number_of_nodes())
print('Cantidad de aristas:',G.number_of_edges())
# Ver nodos y aristas del grafo
print('Nodos: \n',list(G.nodes))   # ver nodos
print('Aristas: \n',list(G.edges)) # ver aristas

# ver adyacencias de los nodos
print('Adyacencias de A: \n',list(G.adj[0])) # ver adyacencias para el nodo A o 0
print('Adyacencias de B: \n',list(G.adj[1])) # ver adyacencias para el nodo B o 1
print('Adyacencias de C: \n',list(G.adj[2])) # ver adyacencias para el nodo C o 2
print('Adyacencias de D: \n',list(G.adj[3])) # ver adyacencias para el nodo D o 3
# Visualizar el grafo
nx.draw(G)  # dibujar grafo
plt.title('Grafo del mapa con 4 distritos',fontsize=16) # titulo
plt.axis('off')  # ocultar ejes
plt.show()  # mostrar dibujo
```

El resultado es el siguiente:

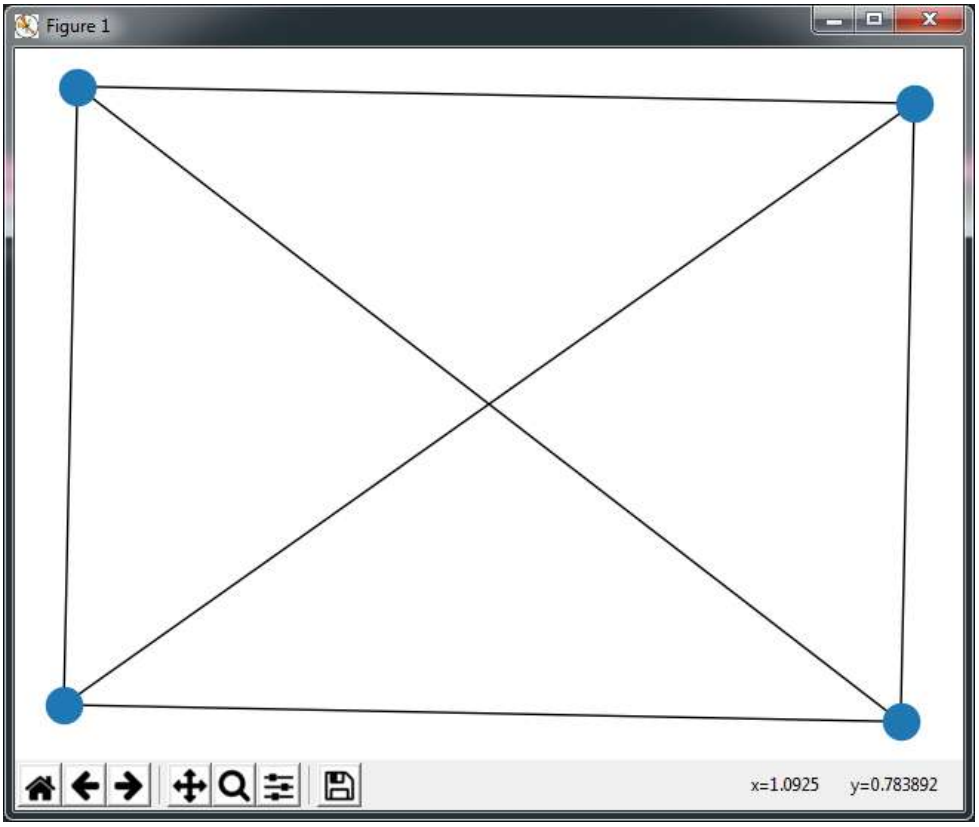

Figura 5. Grafo hecho con Python.

Morgan G. Vásquez
Ingeniero mecatrónico 06/06/2019

BIBLIOGRAFIA

[1] http://www.dma.fi.upm.es/personal/gregorio/grafos/web/iagraph/coloracion.html

[2] https://www.youtube.com/watch?v=moZdSyN1KuI

[3] http://blog.andresed.me/2015/08/algoritmos-de-coloracion-de-grafos.html