

Interfaces Gráficas con Aplicaciones

Alumnos:	Flores Rodríguez Francisco Jacob González Rodríguez Germán Antonio
Profesor:	Dr. Alberto Salvador Núñez Varela

Descripción del problema

El problema por resolver es el conocido como Graph Coloring. En este problema lo que se plantea es que se van a colorear los nodos (en otros casos también los aristas) de un grafo tomando en cuenta 2 factores:

- Los nodos adyacentes deben ser de un color distinto al del nodo principal.
- El número de colores a utilizar deberá ser el mínimo posible.

¿Cómo se resuelve el problema de manera secuencial?

Se puede resolver tomando varios algoritmos como referencia, ya que es un tema muy investigado y que tiene muchas aplicaciones. Está el enfoque recursivo, el backtracking, DSatur, y más, pero en mi caso planteé una solución de fuerza bruta (Brute force) porque no supe implementar los algoritmos mencionados anteriormente.

¿Cómo resolverá el problema de manera concurrente?

La resolución del problema será creando un hilo por cada nodo/vértice que exista en el grafo. De manera que, al final de todo se deberán respetar los puntos en los que consiste el algoritmo.

Resultados

Es complicado hacer n hilos porque los nodos se tendrían que repartir en distintos hilos, por lo que lo hare corriendo tantos hilos como nodos.

En esta sección presenta los resultados en una tabla similar a la siguiente:

Datos	1 hilo (secuencial)	n hilos (El número de nodos)
25 (5x5)	576	428
100 (10x10)	441ms	593
2,500 (50x50)	360	567
10000 (100x100)	336	999
40,000 (200x200)	631	335
25,000 (500x500)	Ya no creó los nodos y relaciones	Ya no me dejó

Análisis

Hacer el programa fue difícil, ya que requería de la comprensión de nuevos conceptos para mí, incluyendo los algoritmos. Aún así, después de pensar como implementar los algoritmos en el programa, pensé en mejor hacerlo yo mismo y dejar de complicarme intentando comprender otros códigos.

Después de hacer mi propio código (que muy probablemente no es eficiente con una mayor cantidad de datos) pude comenzar con la implementación del enfoque con hilos, lo cual fue muy difícil para mí, puesto que me empiezo a confundir y lo implementé tomando como base el enfoque secuencial.

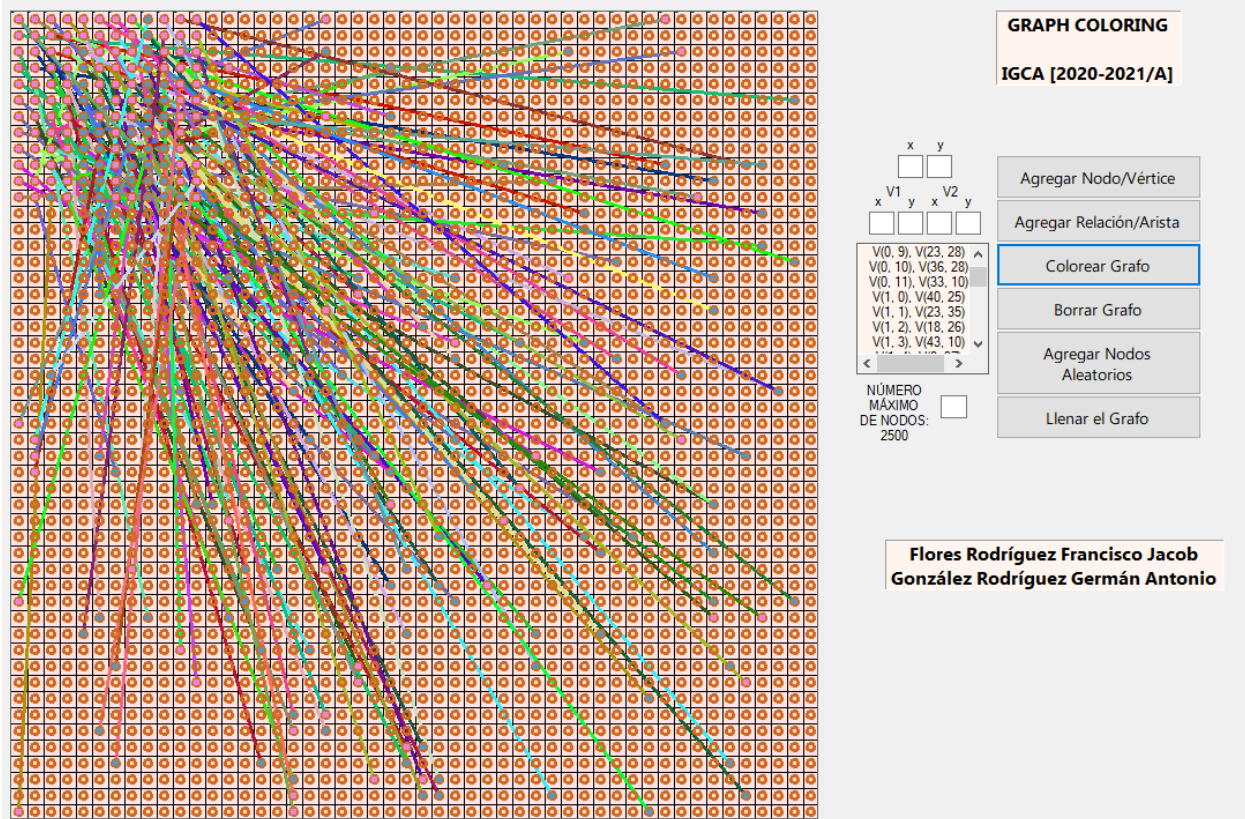
El tiempo que tardan en correr los hilos también dependerá de las relaciones que haya entre estos, ya que puede haber muchas o pocas. Esto sucede porque utilicé la creación de los grafos aleatorios que también generan aristas aleatorias.

El método que utilicé para medir los tiempos fue:

```
Stopwatch time = new Stopwatch();  
time.Start();  
c.GraphColoring(this);  
time.Stop();
```

Además, al estar dibujando los nodos, cuando ya eran muchos datos (a partir de 100x100 tardaba mucho en tan solo llenar la malla).

Cuando el programa tardaba mucho era por el número de relaciones que se generaban automáticamente. Las cambié por $\frac{1}{4}$ de las que eran.



Los colores ya ni siquiera se alcanzan a ver bien.

Al final, con hilos tuvo una mayor velocidad en la última prueba con más datos, pero ninguno de los dos tardaba mucho en el coloreo, aunque en general secuencial fue más rápido, pero esto ha de ser por la forma en que lo implementé que no fue la más óptima. Era mayor el tiempo que se tardaba en hacer la malla, vértices y relaciones.

También debo tomar en cuenta de que puse un `Thread.Sleep(200)`, entonces en eso influiría cada tiempo (o eso supongo) de los hilos. Si no lo ponía no se coloreaban todos los nodos. Además tenía otras apps abiertas, por lo que eso podría haber influido.