

TALLER N°2

Clique máximo de un grafo



Taller de programación 1-2024

Fecha: 31-05-24

Autor: Benjamín Zúñiga Jofré



TALLER N°2

Clique máximo de un grafo

Explicación breve del algoritmo

El programa consiste en encontrar el clique máximo de un grafo, lo cual en otras palabras se podría describir como, encontrar dentro de un conjunto o grupo de vértices unidos entre sí por aristas el subconjunto o subgrupo de vértices donde cada vértice de este subconjunto o subgrupo este unido a todos los demás vértices del subconjunto o subgrupo y que a la vez este subconjunto o subgrupo tenga la mayor cantidad de vértices posibles.

Para lograr encontrar el clique máximo de un grafo se utiliza el algoritmo de Bron-Kerbosh, el cual consiste en ir revisando cada vértice del grafo y ir formando el clique máximo de manera recursiva, para esto el algoritmo usa tres conjuntos, en este caso uno llamado R, otro P y el último X, donde R es el clique que se lleva formado, P que es el conjunto de vértices por revisar y X el conjunto de vértices ya revisados. En palabras simples el algoritmo revisa todos los vértices que faltan por revisar, es decir, los del conjunto P y se agregan al clique máximo actual si es que el vértice está conectado con los demás vértices que forman el clique máximo actual, es decir, los del conjunto R y por último se agregan al conjunto de los ya revisados es decir, al conjunto conjunto X para no volver a ser revisados, más adelante se darán más detalles de cómo funciona el algoritmo en sí.

Heurísticas o técnicas utilizadas

Dentro de las técnicas utilizadas para mejorar los tiempos del programas, encontramos el uso de Greedy, debido a que a la hora de formar el conjunto P de vértices, este no es formado al azar, sino que primeramente se inserta el vértice con más conexiones dentro del grafo, luego los vértices con más conexiones con los vecinos del vértice con más conexiones y por último se insertan los vértices que faltan según sus cantidades de conexiones de mayor a menor, esta técnica funciona la mayoría de veces debido a que el vértice con más conexiones es el que tiene más probabilidad de formar parte del clique máximo, al igual que aquellos vértices que comparten más vecinos con el vértice con mayor cantidad de conexiones. También se utiliza la selección de un pivote a la hora de ir escogiendo vértices a revisar, este pivote es seleccionado de entre la unión del conjunto P y X, es decir, todos los vértices y es seleccionado el vértice con mayor cantidad de conexiones con la intersección de P y los vecinos del vértice, es decir el vértice que tenga más vecinos dentro de P, esto es debido a que luego al escoger un vértice para revisar si podría formar parte del clique, si el vértice elegido es vecino del pivote, no es necesario revisarlo, pues a la hora de revisar el pivote se revisaran todos sus vecinos, ahorrando revisiones de esta manera. Y por último se utilizó poda de recursiones, la cual se hace mediante una cota mínima, la cual es que el tamaño del conjunto S más el del conjunto P es menor al tamaño del clique máximo que se lleva hasta el momento se detiene



esa rama de revisión pues por ese camino nunca se va a llegar a un clique con mayor tamaño que el que se lleva hasta ese momento

Funcionamiento del programa

El programa funciona primeramente creando un objeto de la clase clique mediante la lectura de un archivo el cual tiene un formato específico, partiendo con el número de vértices en la primera línea del archivo y posteriormente todas las líneas son dos números que representan una conexión entre los vértices indicados por cada número, luego se crea una matriz de adyacencia que representa el grafo del archivo y luego se calculan la cantidad de vecinos que tiene cada vértice del grafo y se guardan en un vector para ser utilizado más adelante. Luego de la creación del objeto clase clique se crean los conjuntos R, P y X vacíos y se busca el vértice con más conexiones, para así encontrar los vecinos de ese vértice, agregarlos a un vector y luego modificar la cantidad de vecinos de cada vértice, en base a cuantos vecinos tiene en común con el vértice con más conexiones. Posteriormente se rellena el conjunto P por su cantidad de vecinos totales y se comienza el algoritmo de Bron-Kerbosh para encontrar el clique máximo del grafo. El funcionamiento del algoritmo de Bron-Kerbosh es primero se revisa si el clique actual, es decir, el que se lleva es mayor que el máximo total, el cual se guarda como atributo del objeto, en caso de que si sea mayor se actualiza el atributo, luego si el conjunto P y X son vacíos se termina la búsqueda, ya que esto significaría que se revisaron todos los vértices, después se copia el conjunto P dos veces y el X una vez, para así no tener problemas al insertar o eliminar vértices del conjunto y se obtiene el pivote, luego se revisan todos los vértices que quedan por revisar, es decir, los que están en el conjunto P, posteriormente se revisa si el vértice escogido no es vecino del pivote, en caso de serlo se escoge otro vértice y en caso de no serlo se crean un nuevo conjunto R, P y X, al nuevo conjunto R se le agrega el vértice escogido y en el nuevo P y X se guarda la intersección de los vecinos del vértice con los vértices del conjunto P original y los vértices del conjunto P original, respectivamente, por último si la suma de los tamaños del nuevo conjunto R y P es mayor al clique máximo guardado hasta el momento se llama al algoritmo nuevamente con los nuevos conjuntos R, P y X, sino el vértice escogido se elimina de P y se inserta en X y aparte se eliminan los nuevos conjuntos R, P y X para liberar memoria.

Aspectos de implementación y eficiencia

El programa se podría considerar eficiente tanto en eficiencia, como en utilización de memoria, siempre considerando las pruebas realizadas con unos grafos en particular, pues por el lado de eficiencia el programa no se demora más de un segundo en encontrar el clique máximo en ninguno de los grafos puestos a prueba (Tabla de tiempos en sección anexos). Mientras que por el lado de la memoria a pesar de que se crean copias de los conjuntos estas se eliminan cada vez que no se utilizan, siendo así que no se desperdician recursos de memoria.



Ejecución del código

Para ejecutar el código basta con abrir una terminal del sistema con la dirección de la carpeta que contiene el código del programa y ejecutar el comando “make main” para compilar el programa principal o ejecutar el comando “make testClique” para compilar una prueba de la clase, luego ejecutar “./main” o “./testClique” según corresponda y por último ingresar el nombre del archivo que contiene un grafo a resolver, todo esto considerando que se usa un computador con alguna distribución del sistema operativo Linux.

Anexos

Nombre del archivo	Tiempo [s]	Tamaño clique máximo
clique_1.txt	0.007187	23
clique_2.txt	0.012857	20
clique_3.txt	0.013056	24
clique_4.txt	0.087501	22
clique_5.txt	0.222040	23
clique_6.txt	0.062949	28
clique_7.txt	0.049162	35
clique_8.txt	0.897822	26

Tabla 1: Tiempos de ejecución