TALLER N°2

Problema del Clique Máximo

Taller de programación 1-2024

Fecha: 20 de mayo de 2024

Autor: Vicente Gerardo Arce Palacios

TALLER N°2

Problema del Clique Máximo

Explicación breve del algoritmo

El algoritmo usado para el problema de encontrar un clique máximo, es el algoritmo de Bron-Kerbosch, este algoritmo, si bien de base no es un algoritmo el cual sea demasiado eficiente, con algunas heurísticas y tipos de datos puede ser optimizado. Un clique es un subconjunto de vértices tal que cada par de vértices está conectado por una arista. El algoritmo inicia con tres conjuntos: R (vértices en el clique actual), P (vértices candidatos a añadir) y X (vértices ya procesados). Recursivamente, se añade un vértice de P a R, actualizando P y X con sus vecinos. Si P y X están vacíos, entonces R es una clique máxima.

Heurísticas o técnicas utilizadas

En este algoritmo se hizo uso de la técnica de ramificación y poda en base a la selección de un pivote, en este caso dicho pivote es el nodo que tenga mayor grado. El objetivo es reducir el número de recursiones al evitar la exploración de algunos subconjuntos de vértices. La técnica de pivote se basa en seleccionar un vértice que está conectado con la mayoría de los vértices restantes, lo que ayuda a minimizar el espacio de búsqueda.

Funcionamiento del programa

El algoritmo Bron-Kerbosch implementado sigue los siguientes pasos:

1. Inicialización:
    - Se crean tres conjuntos: R, P y X.
        - R representa el conjunto de vértices que forman el clique actual.
        - P contiene los vértices candidatos a ser incluidos en el clique.
        - X contiene los vértices que ya se han considerado y no pueden formar parte del clique actual.
2. Condición de Parada:
    - Si P y X están vacíos, significa que se ha encontrado un clique. Se compara el tamaño de R con el tamaño del clique máximo encontrado hasta el momento y se actualiza si es necesario.
3. Selección de Pivote:
    - Se selecciona un pivote utilizando la función getOptimalPivot, que elige el vértice con la mayor cantidad de vecinos en P y

X. Este paso ayuda a reducir la cantidad de ramas a explorar, optimizando el algoritmo.

4. Iteración sobre los Vértices en P:
    - Para cada vértice vertex en P que no sea vecino del pivote:
        - Se crea una copia del conjunto R (Rcopy) y se añade el vértice actual.
        - Se crean dos nuevos conjuntos, Pnew y Xnew:
            - Pnew contiene los vecinos del vértice actual que están en P.
            - Xnew contiene los vecinos del vértice actual que están en X.
        - Se llama recursivamente a bronKerbosch con Rcopy, Pnew y Xnew, siempre y cuando la suma de los tamaños de Rcopy y Pnew sea mayor a el tamaño del clique máximo actual.
5. Actualización de Conjuntos:
    - Después de explorar un vértice, se elimina de P y se añade a X para evitar que sea reconsiderado en futuras ramas de la búsqueda.

Aspectos de implementación y eficiencia

El programa se enfoca en lograr optimizar de gran manera el algoritmo de Bron-Kerbosch, esto a través de la heurística ya mencionada que consta en elegir como pivote aquel nodo con mayor grado de los conjuntos P y X, esto con la finalidad de podar aquellos casos que sean más desfavorables. El programa a su vez se considera eficiente ya que, con los archivos proporcionados por el profesor, todos los cliques lograron encontrarse en menos de 0.5 segundos, tal y como se muestran los resultados en la siguiente tabla:

Nombre de Archivo	Tiempo [segundos]	Tamaño del clique
Clique1.txt	0.006125	23
Clique2.txt	0.009874	20
Clique3.txt	0.012481	24
Clique4.txt	0.059787	22
Clique5.txt	0.114637	23
Clique6.txt	0.018566	28
Clique7.txt	0.24134	26
Clique8.txt	0.046851	35

Ejecución del código

Para poder ejecutar de manera correcta el programa, se necesita operar en un entorno con el sistema operativo Linux, para prevenir errores o fallos, este programa fue escrito y probado en Ubuntu 23.10. Además, debe asegurarse de que todos los archivos .h y .cpp se encuentren en un mismo directorio/carpeta, así mismo los archivos .txt de los grafos deben estar organizados en el mismo directorio. A continuación, se describe el proceso que se debe seguir para la correcta ejecución del programa:

1. Abrir la terminal: Abra una terminal en Linux y navegue hasta el directorio que contiene todos los archivos mencionados con anterioridad (carpeta “Codigo”). Si quiere ahorrarse el paso de ir navegando hasta llegar al directorio, simplemente abra este y ejecute una nueva terminal.
2. Compilación [makefile]: Ejecute el comando make en la terminal para que automáticamente se compilen los archivos necesarios para la correcta ejecución del programa.
3. Ejecución:
    a. En la misma terminal, si desea ejecutar el programa introduzca “./main”, esto hará que el programa main se ejecute.
    b. El programa inicia con un menú el cual le dará dos opciones, la primera (“Encontrar clique máximo”) le pedirá proporcionar el nombre del archivo, por ejemplo “clique4.txt”, al momento de presionar la tecla “enter” el programa ejecutara el algoritmo que le proporcionara el tamaño y los vértices que conforman el clique máximo del grafo. La segunda opción que se le da en el menú es la de salir, que como su nombre lo indica, se termina la ejecución.
    c. Importante: si los archivos no se encuentran en el mismo directorio es posible que la compilación del programa no se realice con éxito, a su vez si el archivo que contiene los datos del grafo se escribe mal, se pedirá al usuario introducir correctamente el nombre del archivo.
4. Resultados esperados: Si todos los pasos resultaron éxitos, entonces el programa imprimirá el tamaño del clique máximo del archivo que se proporcionó, los nodos que componen dicho clique y el tiempo de ejecución del algoritmo.

Importante: Se asumirá que el archivo proporcionado es correcto, es decir, que contiene los datos necesarios para formar el grafo no dirigido.

Bibliografía

González, F. (2021). *Tesis_Francisco González domínguez_abril 2021.PDF*. Algoritmo de enumeración de cliques maximales en grafos k-outerplanares, usando descomposición de árbol.https://cicese.repositorioinstitucional.mx/jspui/bitstream/1007/3559/3/tesis_Francisco González Domínguez_abril 2021.pdf

Anexos

"Anexo 1: Ejemplo de ejecución clique1.txt"

```
##### Menu #####
1) Encontrar clique máximo
2) Salir
Ingrese una opción: 1
Introduza el nombre del archivo: clique1.txt
0 1 1 1 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1
1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1
1 1 0 0 1 0 1 1 1 1 0 0 1 1 1 1 1 0 0 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1
1 1 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1
1 0 0 1 1 0 1 1 0 1 1 1 1 1 0 0 1 0 1 1 0 1 1 0 0 1 0 1 1 1 0 0 0 1 0 0 1 0 0 0 1 0 1 1 0 1 0 1 1 0
0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 0 0 0 1 1 0 0 1 1 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 1 1 1
1 1 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1 1
0 0 1 0 0 0 1 0 0 1 0 0 0 1 1 1 1 0 1 1 0 1 1 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 0 0 1 0 1 0 1
1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 1
1 1 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1
0 1 0 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1
1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 0 1 0 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 1 0 1 0 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1
0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 1 1 1 1 1
0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 0 1 1 0 1 1 1
1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 0 0 0
0 0 0 1 0 1 0 0 1 1 0 1 0 1 0 1 0 0 0 1 1 0 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
1 1 0 1 1 0 0 0 0 1 0 1 0 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1
1 1 1 0 1 1 1 1 1 1 0 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1
1 1 1 1 0 1 1 0 1 1 1 0 1 1 0 0 1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0
1 1 1 1 1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1
0 1 0 1 1 0 0 0 0 0 0 1 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 1 1 1 0 1 0 0 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1
1 0 0 1 0 1 0 1 0 1 1 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 0 0 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
0 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1
1 1 0 1 0 0 0 1 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 1 0 1 0 0 1 1 1 0 0 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 0 1 1 0 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1
0 0 1 1 0 0 1 0 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 0
1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 0 1 1 0 1 1
1 1 1 1 1 0 0 1 0 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 1 0 1 0 0 0 0 1 1 1 0 0 0 1 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 0 1 0 1 0 1 0
0 0 0 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 0 0 1 1 1 1 0 1 1 1 0 0 1 0 1 1
1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 0 0 0 1 0 1 1 1 1 0 1 1 0 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 1 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1
1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0
Clique maximo: {1, 2, 4, 7, 13, 14, 15, 19, 23, 25, 26, 29, 30, 33, 34, 35, 36, 39, 41, 42, 47, 48, 49}
Tamaño: 23
Tiempo: 0.005949 segundos
```