Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-2693 - Laboratorio de Algoritmos y Estructuras III
Trimestre Septiembre-Diciembre 2021

Árbol mínimo cobertor y componentes conexas

1. Introducción

El primer objetivo del laboratorio es la implementación de los algoritmos de Prim y Kruskal, para obtener el árbol mínimo cobertor de un grafo no dirigido. El segundo objetivo es la implementación de dos métodos de obtención de componentes conexas en grafos no dirigidos. El primer método es por medio de estructuras de conjuntos disjuntos. El segundo método se basa en búsqueda en profundidad.

2. Actividades a realizar

2.1. Algoritmos de componentes conexas y árbol mínimo cobertor

Se quiere agregar a la librería *grafoLib* las siguientes clases:

ConjuntosDisjuntos: Estructuras para conjuntos disjuntos, usando la representación de árboles.

ComponentesConexasCD: Detección de componentes conexas haciendo uso de estructuras para conjuntos disjuntos.

ComponentesConexasDFS: Detección de componentes conexas haciendo uso de la búsqueda en profundidad.

ArbolMinimoCobertorKruskal: Determina el árbol mínimo cobertor usando el algoritmo de Kruskal. El algoritmo de Kruskal debe hacer uso de las estructuras para conjuntos disjuntos.

ArbolMinimoCobertorPrim: Determina el árbol mínimo cobertor usando el algoritmo de Prim. El algoritmo de Prim debe hacer uso de heaps binarios.

Se le proporcionará de un código base, contenido en el archivo `codigoBaseLabSem5.tar.xz`. Este código contiene los archivos adicionales que usted debe agregar a la librería *grafoLib*. Se debe completar y documentar el código de las actividades a realizar. Puede hacer uso de las clases de la librería de Kotlin para su implementación. Cada una de las operaciones en las clases tiene una breve descripción la misma. Esa descripción debe ser borrada de su código de entrega. En su lugar deben colocar para cada una documentación de las operaciones en las que se indique *descripción*, *precondiciones*, *postcondiciones* y *tiempo de la operación*. El tiempo de las operaciones debe ser dado en número de lados y/o número de

vértices, cuando eso sea posible. Sus implementaciones deber ser razonablemente eficientes. Debe entregar la librería *grafoLib* completa, con todos los códigos de las implementaciones de esta semana y de las semanas anteriores, junto con un archivo makefile, llamado `Makefile`, que compila solamente a la librería. Las implementaciones de sus soluciones deben estar basadas en los pseudo códigos de los algoritmos vistos en clase y en [1].

2.2. Pruebas experimentales

Se quiere que realice dos pruebas experimentales. La primera prueba tiene como objetivo comparar a los algoritmos para la detección de componentes conexas en grafos no dirigidos. Para ello debe crear un programa cliente llamado `PruebaCC.kt`. Este programa recibe como entrada un grafo no dirigido y muestra como salida el tiempo que le tomó a la clase ComponentesConexasCD y a la clase ComponentesConexasDFS obtener las componentes conexas. También se debe mostrar por la salida estándar el número de componentes conexas del grafo de entrada. El cliente `PruebaCC.kt` debe ser ejecutado en un archivo ejecutable llamado `pruebaCC.sh` que recibe como entrada un archivo que contiene un grafo no dirigido, con el formato dado en clase.

Con la segunda prueba se quiere comparar el tiempo que le toma a las implementaciones de los algoritmos de Prim y Kruskal obtener las componentes conexas. Debe crear un programa cliente llamado `PruebaAMC.kt` que recibe como entrada un grafo dirigido y muestre por la salida estándar el tiempo que le tomo a la clase ArbolMinimoCobertorKruskal obtener el árbol mínimo cobertor, y de la misma manera a la clase ArbolMinimoCobertorPrim. También se debe mostrar por la salida estándar el peso del árbol mínimo cobertor. El programa cliente debe poder ser ejecutado por medio de un *shell script* llamado `pruebaAMC.sh` que recibe como entrada un archivo que contiene un grafo no dirigido, con el formato dado en clase. Se debe crear un archivo makefile, llamado `Makefile.cliente` que sea capaz de compilar el código de estos programas clientes y el de la librería *grafoLib*.

Se le va a proporcionar de dos conjuntos de grafos de prueba. Para probar los algoritmos de detección de componentes conexas se le proporcionará de tres archivos con grafos no dirigidos. Los archivos están contenidos en un archivo llamado `grafosComponentesConexas.tar.xz`. El segundo conjunto de grafos de prueba son para evaluar los algoritmos que encuentran el árbol mínimo cobertor. Los archivos con los grafos están contenidos en el archivo `grafosAMC.tar.xz`.

Finalmente debe hacer un reporte con los resultados experimentales de la evaluación de los algoritmos de detección de componentes conexas y de obtención de árbol mínimo cobertor. En el reporte se debe indicar las especificaciones del equipo en donde se realizaron las pruebas. Esto es, sistema de operación, CPU, memoria RAM y versión de Kotlin utilizada. El reporte debe contener cuatro tablas. La primera tabla muestra para cada grafo evaluado, el número de componentes conexas que tiene. La segunda tabla muestra el tiempo de ejecución de los algoritmos de detección de componentes conexas para cada uno de los grafos ejecutados. La tercera tabla muestra el peso del árbol mínimo cobertor de cada uno de los grafos estudiados. La cuarta y última tabla contiene los tiempos de ejecución de los algoritmos de Prim y Kruskal para cada uno de los grafos de evaluación. Finalmente el reporte debe tener dos secciones de análisis de resultados. En la primera

sección se analiza los resultados de los algoritmos de detección de componentes conexas. En la segunda sección debe discutir los resultados de los algoritmos de Prim y Kruskal. El informe debe estar en formato PDF.

3. Condiciones de entrega

Los códigos del laboratorio, el informe y la declaración de autenticidad debidamente firmada, deben estar contenidos en un archivo comprimido, con formato *tar.xz*, llamado *LabSem5_X.tar.xz*, donde X es el número de carné del estudiante. La entrega del archivo *LabSem5_X.tar.xz*, debe hacerse por medio de la plataforma *Classroom* antes de las 12:00 pm del día lunes 15 de noviembre de 2021.

Referencias

[1] Cormen, T., Leirserson, C., Rivest, R., and Stein, C. *Introduction to Algorithms*, 3ra ed. McGraw Hill, 2009.

Guillermo Palma / gvpalma@usb.ve / 12 de noviembre de 2021