Universidad de Concepción

Ingeniería Civíl Informática

Estructura de Datos

Proyecto 3

Alexander Irribarra, Angelo Zapata

Profesor Diego Seco
Ayudantes: Diego Gatica, Paulo Olivares

23 de Junio, 2017

Índice

1 Introducción

Una red social es un sistema de relaciones entre individuos de una sociedad o grupo, compuesta por un conjunto de actores, que pueden ser individuos personales u organizaciones, que están relacionados de acuerdo a algún criterio.

El uso de las redes sociales en internet, en la actualidad, es algo muy importante para entablar relaciones entre individuos que compartan algún interés en común o alguna relación en algún ámbito de sus vidas.

Para este proyecto se construirá una mini red social, muy similar al conocido Twitter, en el cuál la relación entre las personas será a qué personas siguen (Follow). Esto se implementará de una forma abstracta con un grafo y con sus correspondientes métodos para que sea tanto eficiente como fácil de entender.

2 Desarrollo

Se construyó un diGraph para representar la red social. Cada usuario es representado por un nodo del grafo, y las relaciones son almacenadas como las aristas. El diGraph contiene un vector con los punteros a todos los nodos.

Dentro de cada nodo se tienen en vectores los punteros hacia los nodos que sigue y que lo siguen. Así, se tienen almacenado por separado las aristas de entrada y de salida.

También cada nodo está caracterizado por un string para su nombre y un entero representando su indice dentro del vector del diGraph. Respecto a los métodos de los nodos, solamente tiene los básicos para retornar variables y añadir aristas tanto de entrada como de salida.

2.1 Find(u)

Para elegir la mejor implementación de este método, se comparó una forma lineal, recorriendo todos los nodos del grafo (con una complejidad O(n+m), puesto que usa un DFS) y una implementada con un ADT Map de la STL de C++, dando como mejor resultado la implementación con el ADT Map, el cual la complejidad es de O(log n) puesto que está construido sobre un árbol binario de búsqueda, y se decidió por esta por las garantías de peor caso (en comparación de unsorted map implementado con tablas hash), ya que una red social es de un tamaño considerable y debe funcionar bien en tiempo real.

En el análisis experimental.

	1000	10000
Map	0.001799	0.023217
DFS	0.160831	15.6721

Al llamar a este método, retorna si el usuario u existe en el grafo o no.

2.2 Clique()

Un clique es un subgrafo en que cada vértice está conectado a cada otro vértice del subgrafo, es decir, todos los vértices del subgrafo son adyacentes

formando un grafo completo.

Para implementar este método, se uso una solución conocida por Bron-Kerbosch para grafos no dirigidos, pero adaptada a grafos dirigidos. Esta consiste en un método privado recursivo llamado BK(R,P,X) el cual trabaja con 3 subconjuntos de nodos R, P y X, representados como vectores de booleanos, para los cuales en la primera llamada P es el conjunto de todos los nodos, R y X son conjuntos vacíos. Cuando en una llamada P y X son conjuntos vacios, el conjunto R es un clique. Así P y X son conjuntos disjuntos cuya unión son aquellos vértices que forman clique cuando son añadidos a R.

Al llamar a este método, nos imprime los nodos de los cliques.

2.3 Compact()

Para implementar este método, se usó de base el clique descrito anteriormente para obtener los nodos a compactar. Después, solo se consideran cliques disjuntos pues se marcan los nodos que ya están en un clique anterior. Si el clique sin los nodos anteriores sigue sirviendo, es decir sigue siendo de tamaño mayor a 2, se considera para construir un nodo virtual y se le da un nombre e índice para ser mapeado a una matriz de adyacencia. Una vez que se han repasado todos los cliques, se continua con los nodos que no quedaron marcados y se sigue mapeando de la misma forma dándoles nombre e índice.

En este punto ya se tiene mapeado cada nodo del grafo original a uno del grafo compacto. Luego se crea una matriz de adyacencia para el nuevo grafo y se marcan las aristas del grafo original dentro de la nueva matriz, considerando los mapeos. Finalmente solo queda recorrer la matriz imprimiendo las aristas.

Al llamar a este método, compacta los cliques en nodos virtuales compactos e imprime las aristas del grafo resultante.

2.4 Follow(n)

Éste método sugiere los n usuarios más importantes de la red. Para obtener esto, se recorren los nodos y se almacenan en una Priority Queue por el

criterio más importante, que es el mayor grado de llegada del nodo (inDeg), luego por el menor grado de salida (outDeg) y finalmente por orden alfabético.

Al llamar a este método, nos imprime los n usuarios más importantes (o más populares).

3 Archivos

3.1 Node.h y Node.cpp

Contienen respectivamente la definición e implementación de los nodos.

3.2 diGraphADT.h

Contiene la definición del tipo de dato abstracto diGraph y sus métodos.

3.3 diGraph.h y diGraph.cpp

Contienen respectivamente la definición e implementacion de diGraph. Contienen los métodos pedidos. Para el caso de find, las dos versiones están como métodos privados.

3.4 main.cpp

Archivo principal a ejecutarse. Realiza la lectura de entrada y ejecuta los métodos pedidos sobre una instancia de diGraph.

4 Conclusión

Para finalizar el presente proyecto, se puede concluir que una red social es una estructura compleja con una gran cantidad de usuarios conectados entre sí, además de que es posible de comprender a grandes rasgos su funcionamiento y por qué funcionan de forma eficiente.

Para hacerlo mas sencillo se ideó que los grafos son capaces de modelar una red social, representando a los usuarios como los nodos y las relaciones entre ellos con aristas.

En este mini Twitter implementado, se usaron cuatro métodos principales, el find(u) que simplemente nos dice si el usuario existe o no, el clique() que nos dice que usuarios están completamente conectados entre si (podría servir para un posterior análisis, por ejemplo, concluyendo que son compañeros de curso, dado que por lo general se conectan todos con todos), el compact() que le da eficiencia al grafo haciendo más "pequeña" la conexión entre usuarios y por último el follow(n) que nos entrega los usuarios más importantes o populares de la red social (perfectos para hacer alguna difusión).

Todo esto se implementó tanto con algoritmos propios como con otros conocidos, para que sean más eficientes.