

UNIVERSIDAD DE CONCEPCIÓN

INGENIERÍA CIVIL 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	2
2	Desarrollo	3
2.1	Find(u)	3
2.2	Clique()	3
2.3	Compact()	4
2.4	Follow(n)	4
3	Conclusión	5

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 eficiente y fácil de entender.

2 Desarrollo

Se construyó un diGraphADT para representar la red social con los siguientes métodos importantes, con su respectiva explicación de cómo se implementó.

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 es un árbol binario de búsqueda y se decidió por esa implementación por la garantía de peor caso (en comparación de unsorted map implementado con tablas hash), ya que una red social es de un tamaño considerable.

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 llamado $BK(R,P,X)$ el cual trabaja con 3 subconjuntos de nodos R , P y X los cuales en la primera llamada P es el

conjunto de nodos, R y X son conjuntos vacíos, luego se itera entre ellos y cuando P y X son conjuntos disjuntos, el 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 maximales.

2.3 Compact()

Al 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, se considera para hacer un nodo virtual y se le da un nombre e índice para una matriz de adyacencia, una vez que se repasa todos los cliques, 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 original a uno del grafo compacto, entonces se crea una matriz de adyacencia y se marcan las aristas del grafo original a la con los nodos transformados, luego solo se recorre la matriz imprimiendo los pares.

Al llamar a este método, compacta los cliques en nodos virtuales compactos.

2.4 Follow(n)

Este 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 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 fácil de comprender a grandes rasgos su funcionamiento y funcionan de forma rápida, por lo cual son tan populares.

Para hacerlo mas sencillo se ideó que los grafos son capaces de modelar una red social conectada, con los nodos como los usuarios y las arista las conexiones entre ellos.

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 con otros (podría servir como posterior análisis, concluyendo si son compañeros de curso que por lo general se conectan todos con todos), el `compact()` que le da eficiencia al grafo haciéndolo más "pequeño" la conexión entre usuarios y por último el `follow(n)` que nos entregan 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.