**ALGUNAS NOTAS SOBRE GRAFOS**

LIBRO: DATA STRUCTURE AND ALGORITHMS

CAPÍTULO 7.

Muchos problemas en Ciencias de la Computación y Matemáticas pueden reducirse a un conjunto de estados y un conjunto de transiciones entre estos estados. Un grafo es una representación matemática de problemas como estos. En el último capítulo vimos que los árboles sirven para una variedad de propósitos en Ciencias de la Computación. Los árboles son grafos. Sin embargo, los gráficos son más generales que los árboles. Abstraer los detalles de un problema y estudiarlo en su forma más simple a menudo conduce a una nueva comprensión. Como resultado, muchos algoritmos han surgido de la investigación en teoría de grafos. La teoría de grafos fue estudiada por primera vez por matemáticos. Muchos de los algoritmos de la teoría de grafos llevan el nombre del matemático que los desarrolló o descubrió. Dijkstra y Kruskal son dos de esos matemáticos y este capítulo cubre los algoritmos desarrollados por ellos.

Representar un grafo se puede hacer de varias maneras diferentes. La forma correcta de representar un gráfico depende del algoritmo que se implemente. Los problemas de teoría de grafos incluyen colorear grafos, encontrar un camino entre dos estados o nodos en un grafo, o encontrar el camino más corto a través de un grafo, entre muchos otros. Hay muchos algoritmos que han surgido del estudio de grafos. Para comprender la formulación de estos problemas es bueno aprender un poco de notación gráfica que también se presenta en este capítulo.

7.1 Objetivos del capítulo

Este capítulo cubre la representación de grafos. También cubre algunos algoritmos gráficos. Se presenta la búsqueda primero en profundidad de un gráfico, junto con la búsqueda primero en amplitud. El algoritmo de Dijkstra es famoso en Ciencias de la Computación y tiene muchas aplicaciones, desde redes hasta planificación de la construcción. El algoritmo de Kruskal es otro algoritmo famoso que se utiliza para encontrar un árbol de expansión ponderado mínimo. Al final del capítulo, debe tener una comprensión básica de la teoría de grafos y cuántos problemas en Ciencias de la Computación se pueden plantear en forma de gráficos.

Para comenzar, estudiaremos algo de notación y la búsqueda profunda de un gráfico. Luego, examinaremos un par de algoritmos codiciosos que responden algunas preguntas interesantes sobre gráficos. Los algoritmos codiciosos son algoritmos que nunca hacen una elección equivocada al encontrar una solución. Examinaremos dos de estos algoritmos llamados Algoritmo de Kruskal y Algoritmo de Dijkstra, ambos llamados así por las personas que formularon el algoritmo para resolver sus respectivos problemas.

7.2 Notación gráfica

Un poco de notación ayudará en las definiciones de los grafos en este capítulo. Un conjunto es una colección desordenada de elementos. Por ejemplo, V = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} es el conjunto de los primeros 13 números naturales. Un subconjunto de un conjunto es una colección, posiblemente vacía, de elementos de su superconjunto. El conjunto U = {5, 8, 2} es un subconjunto de V. La cardinalidad de un conjunto es su tamaño o número de elementos. La cardinalidad del conjunto V se escribe |V|. La cardinalidad de V es 13 y U es 3, entonces |V| = 13 y |U| = 3. Un gráfo G=(V,E) está definido por un conjunto de vértices, denominados V, y un conjunto de aristas, denominado E. El conjunto de aristas son subconjuntos de V donde cada miembro de E tiene cardinalidad 2. En otras palabras, los bordes se denotan por pares de vértices. Considere el gráfico simple no dirigido de la figura 7.1. Los conjuntos V = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} y E = {{0, 1},{0, 3},{0, 10},{1, 10},{1, 4},{2, 3},{2, 8},{2, 6},{3, 9},{5, 4},{5, 12} ,{5,7},{11, 12},{11, 10},{9, 10}} definen este gráfico. Dado que cada arista es en sí misma un conjunto de cardinalidad 2, el orden de los vértices en cada conjunto de aristas no importa. Por ejemplo, {1, 4} es la misma arista que {4, 1}.

Muchos problemas se pueden formular en términos de un gráfico. Por ejemplo, podríamos preguntar cuántos colores se necesitarían para colorear un mapa para que no haya dos países que comparten una frontera con el mismo color. En este problema, los vértices de la figura 7.1 representarían países y dos países que comparten una frontera tendrían una arista entre ellos. Luego, el problema se puede reformular como encontrar la cantidad mínima de colores necesarios para colorear cada vértice en el gráfico de modo que no haya dos vértices que compartan un borde que tengan el mismo color.

Un grafo dirigido G = (V,E) se define de la misma forma que un grafo no dirigido excepto que el conjunto de aristas, E, es un conjunto de tuplas en lugar de subconjuntos. Al definir E = {(vi, vj) donde vi, vj V} significa que los bordes se pueden atravesar en una dirección solamente. En la Fig. 7.2 podemos movernos del vértice 10 al vértice 0 a lo largo de la arista (10,0), pero no podemos movernos del vértice 0 al 10, al menos no sin pasar por otros vértices, porque la arista (0,10) no está en el conjunto E.

Un camino en un gráfico es una serie de aristas, ninguna repetida, que se puede recorrer para viajar de un vértice a otro en un gráfico. Un ciclo en un gráfico es un camino que comienza y termina con el mismo vértice. El último capítulo cubrió los árboles en informática. Ahora, armados con alguna notación de la teoría de grafos, podemos dar una definición formal de un árbol. Un árbol es un grafo acíclico dirigido y conexo. Un gráfico acíclico es un gráfico sin ningún ciclo.

A veces, en la teoría de grafos, un árbol se define como un gráfico conectado acíclico, eliminando el requisito de que sea un gráfico dirigido. En este caso, un árbol puede definirse como un grafo que está completamente conectado, pero tiene solo un camino entre dos vértices cualesquiera.

Tanto los gráficos dirigidos como los no dirigidos se pueden usar para modelar muchos tipos diferentes de problemas. El gráfico de la figura 7.1 podría representar la asignación de registros en una CPU. Los vértices podrían representar registros con nombres simbólicos y dos registros que estaban en uso al mismo tiempo tendrían un borde entre ellos. La pregunta que podría hacerse es, "¿Cuántos registros físicos de la máquina se requieren para los registros simbólicos de este cálculo?".