Graphs

Lisandro Martinez

March 19, 2020

1 Conceptos

1.1 Grafo

1. Se utilizan para modelar relaciones.
2. Grafo no dirigido (*undirected graph*) $G = (V, E)$
3. V = conjunto de nodos (modelan entidades)
4. E = conjunto de pares de nodos (modelan relaciones)
5. Tamaño:
   (a) $n = |V|$: Número de nodos.
   (b) $m = |E|$: Número de relaciones.
6. Grafo Completo: tiene todas las aristas posibles.

$$\binom{n}{2} \in \mathcal{O}(n^2)$$

7. nodos vecinos: nodos adyacentes.

1.2 Representación

1.2.1 Matriz de adyacencia

- Matriz de $n \times n$.
- $A_{uv} = 1$ si (uv) es una arista.

- Es una matríz simétrica.
- No se puede tener una arista a sí mismo.
- Dos representaciones para cada arista.
- Espacio N^2.
- Verificar si una arista existe $\mathcal{O}(1)$.
- Identificar si una arista existe $\mathcal{O}(N^2)$ (hay que revisar la mitad de la matríz).

Lo bueno es que se puede verificar rapidamente si existe una arista.
Lo malo es que ocupa mucho memoria

1.2.2 Lista de adyacencia

- Por cada nodo se tiene una lista con las aristas.
- Dos representaciones por cada arista.
- Espacio proporcional a $m + n$.
  Si el grafo tiene muchas aristas, lo que va a dominar la cantidad de memoria va a depender de las aristas. En el caso contrario, está dominada por la cantidad de nodos. La complejidad la va a dominar la variable que crezca más rápido. Se necesita memoria proporcional al número de nodo.
- Verificar si existe una arista (u, v) toma $\mathcal{O}(\deg(u))$
  Toma el grado del nodo, es decir, el número de aristas. Hay que recorrer toda la lista en el caso de que sea el último nodo.
  La cota superior es $\mathcal{O}(n)$
- Identificar todas las aristas toma $\Theta(m + n)$

1.2.3 Utilizacion

- Si el grafo es *sparse* (disperso) tiene pocas aristas, la matríz de adyacencias consumiría mucho espacio.
- trade-off memoria por procesamiento.

2 Camino y Conectividad

2.1 Camino

- Def: Un camino en una grafo dirigido es una secuencia P de nodos $v_1, v_2, v_2, \ldots, v_{k-1}$
  - Propiedad: para cualquier par v_i, v_{i+1} adyacente en la secuencia debe haber una arista que los conecte.
- Camino Simple: todos los nodos son distintos. Se pasa una sola vez por cada nodo.
- Ciclo: camino que nos regresa a un mismo lugar.

2.2 Conectividad

- Def. Un grafo no dirigio está *conectado* si para cada par de nodos u y v, existe una camino entre u y v que los une.

3 Inducción matemática

Theorem 3.1. *Dados $S(n)$ y un predicado en $n \in -\mathbb{Z}^+$ Si:*

1. *$S(1)$ es verdadero*
2. *Para cualquiera $S(k)$ con $k \in -\mathbb{Z}^+$ arbitrario, entonces $S(k+1)$ es verdadero.*

Entonces, $S(n)$ es verdadero para todos $n \in -\mathbb{Z}^+$

4 Árboles

4.1 Propiedades

Theorem 4.1. *Dado un grafo no dirigido G, las siguientes definiciones son equivalentes. La anterior implica la que sigue.*

1. *G es un árbol: está conectado y no contiene ciclos.*
   Si un grafo está conectado y no contiene ciclos, se cumple la siguiente definición.
2. *Cualquiera dos vértices en G están conectados por un único camino simple.*
   Tiene dos partes, la primera dice que está conectada y la segunda que está conectada por un camino simple. Se supone que se niega que está conectada por un camino

simple. Se encuentra conectado por más de un camino. Se debe llegar a que debe tener un ciclo. Tomando dos nodos u y v que están conectados por más de un camino P y P', se contradice con 1 (que dice que no tiene ciclos)

3. *G está conectado, pero si se remueve una arista de E, el grafo resultante está desconectado.*

   Se supone que el grafo está conectado por un camino simple. Dados dos nosos u y v conectados por una arista. Si suponemos que caulquier para de nodos está conectado por un camino simple, no hay otra forma de llegar de u a v. Si se remueve la arista, el grafo está desconectado.

4. *G está conectado y $|E| + 1 = |V|$.*

   Proof. (a) inducción sobre el número de aristas.

   (b) Si hay una sola arista, hay dos nodos (una arista más uno). Un nodo es igual a 0 más uno.

   (c) Hipótesis de inducción: se supone que el predicado es válido para k aristas. El predicado es que la relación entre nodos y aristas es $|E| + 1 = |V|$. Se supone que está conectado y si se quita cualquier arista se desconecta el grafo.

   (d) Se toma cualquier grafo de $k + 1$ aristas. Por la Hipótesis se que si se quita una arista se desconecta y quedan $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ donde $0 \leq |E_1|, |E_2| \leq k$ Lo que queda de los dos lados debe tener k o menos de k aristas.

   (e) se aplica inducción fuerte.

$$
\begin{aligned}
|V| &= |V_1| + |V_2| \\
|V| &= (|E_1| + 1) + (|E_2| + 1) \\
|V| &= (|E_1| + |E_2| + 1) + 1 \\
|V| &= |E| + 1
\end{aligned}
$$

□

5. *G es acíclico y $|E| + 1 = |V|$.*

   Se niega que es acíclico. Se supone que es cíclico, que está conectado y hay que contradecir la relación entre nodos y arístas. Se supone que tiene un ciclo de longitud k, con k nodos. El ciclo G_k tiene k nodos y k aristas $G_k = (V_k, E_k), |V_k| = |E_k|$. Que el grafo sea conectado quiere decir que desde cualquier nodo puedo llegar a cualquier otro. Se procede agregando un nodo al grafo, creando el grafo $G_{k+1} =$

$(V_{k+1}, E_{k+1}), |V_{k+1}| = |E_{k+1}|$. Cada vez que se añade un nodo se añade una arista. Se puede seguir haciendo hasta que $Vk + n = V$, de modo que $|V| = |E|$, contradiciendo la Hipótesis. Para demostrarlo fue clave que en un ciclo el número de nodos es igual al número de aristas

6. *G es acíclico, pero si se agrega una arista a E, el grafo resultante contiene un ciclo.* Primero se intenta demostrarlo que $|E|+1 = |V|$ implica que el grafo está conectado. Si se le agrega una arista, se tiene que formar un ciclo. Se supone que existen un montón de pedazos que no están conectados $G_1 \dots G_k$, en los que se cumple la relación entre nodos y aristas.
   $G_1 \cup \cdots \cup G_k$
   $(|E_1| + 1) + \cdots + (|E_k| + 1) = |V_1| + \cdots + |v_k|$
   $|E_1| \cdots |E_k| + k = |V|$
   $|E| + k = |V|$
   $k = S$

4.2 Definiciones

- Árbol enraizado *(Rooted Tree)*: Dado un árbol T, se escoge un nodo raíz r. Al darle una raíz se induce una estructura jerárquica. Existe un padre, hijo, hermanos, ancestros y descendientes, hojas.

5 Conectividad

1. problema de conectividad: Dados dos nodos s y t se desea saber si existe un camino entre s y t.
2. problema del camino más corto: Dados dos nodos s y t se desea saber cuál es el largo del camino más corto entre s y t.

5.1 Algoritmos

5.1.1 BFS - Búsqueda en amplitud

- BFS (búsqueda en amplitud): se comienza en un nodo s y se explora en todas las direcciones posibles, agregando nodos de adyacentes de la siguiente capa.
- Descripción
  - $L_0 = \{s\}$

- L_1 = todos los vecinos de la capa L_0
- L_2 = todos los nodos que no pertenecen a las capas L_0 o L_1 (que no han sido visitados) y que tienen una arista con un nodo en la capa L_1
- L_{i+1} = todos los nodos que no pertenecen a una capa anterior (no hayan sido visitados) y que tienen una arista con un nodo en la capa L_i
-

Theorem 5.1. *Para cada capa j, todos los nodos que estén en la capa L_j están a la distancia exactamente j de s. (se puede usar para calcular el camino más corto)*

Theorem 5.2. *Existe un camino desde s hasta t si y solo si t aparece en alguna capa.*

- Propiedad: Dado un árbol BFS T del grafo $G = (V, E)$(se añaden las aristas cuando se visita el nodo por primera vez. La búsqueda por amplitud lo que hace es podar aristas.). Sean (x, y) parte de G, los niveles de x y x difieren como mucho en 1. Es decir, o quedan en la misma capa o quedan en capas adyacentes.

5.2 Componentes conectados

5.2.1 Definición

Def. Encontrar todos los nodos alcanzables desde s.

5.2.2 Algoritmo

1. Se desea detectar el componente conectado partiendo de un nodo arbitrario s.
2. R consiste en todos los nodos hacia los cuales s tiene un camino.
3. Inicialmente $R = \{s\}$
4. Mientras haya una arista, tal que un nodo esté adentro $u \in R$ y otro esté afuera $v \notin R$, vamos a visitar ese nodo, agregándolo al componente conectado.

Theorem 5.3. *Una vez que el algoritmo termina, R es el componente conectado que contiene s*

Proof. (por contradicción)

- Para cualquier nodo $v \in R$, existe un camino desde cualquier otro nodo $\in R$.

- Dado un nodo $w \notin R$, se da por supuesto que existe un camino $s - w$ P en G.
- Debe existir un primero nodo $v \in P, v \notin R$
- Existe un nodo u que precede inmediatamente a v en P, de modo tal que $(u, v) \in E$
- $u \in R$, lo que contradice la condición de terminación.

□

5.3 DFS - Búsqueda en Profundidad

5.3.1 Algoritmo

1. Marcar el nodo u como visitado y agregar el nodo u a la componente R.
2. Para cada uno de las aristas (u, v) incidentes al nodo u, explorar recursivamente v si no ha sido visitado.

5.4 Propiedades

5.4.1 Propiedad 1

Dada una llamada recursiva al nodo u, todos los nodos que son marcados como visitados desde que llegue al nodo u, todos los nodos desde la invocación de la llamada recursiva hasta su retorno van a ser sus descendientes en el árbol de búsqueda en profundidad T.

Se podan las aristas y se mantienen las aristas mínimas para conectar el gráfo, las cuales generan un árbol.

Theorem 5.4. *Sea T un árbol de búsqueda en profundidad, $x, y \in T$ y $(x, y) \in E$, $(x, y) \in G$, $(x, y) \notin T$. Si me fijo en una arista que estaba en el grafo pero no apareció en el árbol de búsqueda, entonces debe ser cierto que alguno de los dos es el ancestro del otro.*

Proof.
- Se supone que el nodo x es el primero que se visita.
- Cuándo (x, y) es examinado, no se lo agrega a T por que ya fue visitado.
- Dado que y no estaba marcado como visitado cuando se invocó por primera vez *DFS(x)*, y fue descubierto durante la invocación de *DFS(X)* y el final de la llamada recursiva *DFS(X)*.

□

6 Grafo bipartita

6.1 Definición