Grafos No Dirigidos

Definición de Grafo No Dirigido

Un grafo NO dirigido G = (V, A) es lo mismo que un grafo dirigido, excepto que cada arista en A no es un par ordenado de vértices. (v, w) = (w, v)
- Una arista (v,w) es incidente sobre v y w. Los vértices v y w son adyacentes.
- Un camino es una secuencia de vértices tal que cada par consecutivo está conectado por una arista.
- Camino simple: todos los vértices son distintos (excepto que el primero y el último pueden ser iguales).
- Un grafo es conexo si hay un camino entre cada par de vértices.
- Un ciclo simple es un camino simple de longitud >= 3 que empieza y termina en el mismo vértice.

Subgrafos y Árboles Libres

- Un subgrafo G' = (V', A') cumple que V'⊆V y A'⊆A, donde cada arista de A' conecta vértices de V'. Está contenido dentro.
- Subgrafo inducido: incluye todos los vértices V' y todas las aristas entre ellos.
- Árbol libre: grafo no dirigido, conexo y sin ciclos.
- Un árbol libre con n vértices tiene exactamente n-1 aristas.
- Agregar una arista a un árbol libre forma un ciclo.

Representación de Grafos No Dirigidos

- Igual que grafos dirigidos: matrices de adyacencia o listas de adyacencia.
- Las aristas no dirigidas se representan con dos aristas dirigidas (v → w y w → v) son dobles.
- La matriz de adyacencias de un grafo no dirigido es simétrica pero ocupa el mismo espacio

Árboles Abarcadores de Costo Mínimo (AAM)

- Árbol abarcador: árbol que conecta todos los vértices de un grafo sin formar ciclos.
- Tiene |V| - 1 aristas si el grafo tiene |V| vértices.
- Si las aristas tienen costos, el árbol abarcador de menor costo es el árbol abarcador de costo mínimo.
- Propiedad de corte (AAM): si (u,v) es la arista de menor costo entre U y V-U, hay un AAM que incluye (u,v).

Algoritmo de Prim

Algoritmo ávido que crece un árbol a partir de un vértice inicial, agregando siempre la arista de menor costo que conecta U con V-U.
Complejidad: $O(n^2)$

Pseudocódigo:

1. T ← ∅
2. U ← {1}
3. Mientras U ≠ V hacer:
   - Elegir (u,v) de costo mínimo con u en U y v en V-U
   - Agregar (u,v) a T
   - Agregar v a U

Algoritmo de Kruskal

Otro algoritmo ávido que selecciona aristas de menor costo mientras no formen ciclos.
Complejidad: O(a log a)

Pseudocódigo:
1. F ← ∅
2. Mientras F no forma árbol abarcador:
   - Elegir arista de menor costo que no forma ciclo
   - Si une dos componentes distintos, agregarla a F

Búsqueda en Profundidad (DFS)

- Igual que en grafos dirigidos.
- En grafos no dirigidos:
  * Arcos de árbol y de retroceso.
  * Si el grafo es conexo, se obtiene un solo árbol DFS.

Complejidad: O(a)

Pseudocódigo:
1. Comenzar desde vértice v
2. Visitar recursivamente sus adyacentes no visitados
3. Al no tener más adyacentes, retroceder

Búsqueda en Amplitud (BFS)

Explora vértices en orden creciente de distancia desde el origen.
Complejidad: O(a)

Pseudocódigo:
1. Insertar vértice inicial en cola y marcarlo
2. Mientras la cola no esté vacía:
   - Sacar vértice x
   - Para cada adyacente y no visitado:
      - Marcar y y agregar a la cola

Puntos de Articulación y Componentes Biconexos

- Un punto de articulación es un vértice cuya eliminación desconecta el grafo.
- Un grafo es biconexo si no tiene puntos de articulación.
- Un vértice v es punto de articulación si tiene un hijo w tal que bajo[w] ≥ número_bp[v]
- Se usan los valores bajo[] y número_bp[] para detectarlos durante DFS.

Complejidad: O(a) si se usa lista de adyacencia.