Investigación Operativa - Segundo cuatrimestre 2019

Práctica 5: **Algoritmos en grafos**.

1. Indique en cada caso qué tipo de representación sería más adecuada (matriz de adyacencia o lista de vecinos) y cuál es el número de operaciones necesarias:
    a) comprobar si el vértice u es adyacente al vértice v.
    b) calcular el grado del vértice u.
    c) agregar una arista entre los vértices u y v.
    d) eliminar la arista entre los vértices u y v.
    e) calcular el número de aristas del grafo.
    f) comprobar si el grafo es regular.

2. Escriba una modificación del algoritmo BFS para:
    a) detectar si un grafo es bipartito.
    b) calcular la distancia entre dos vértices dados y recuperar un camino mínimo.
    c) calcular la cantidad de caminos mínimos entre dos vértices dados.

3. Diseñe un algoritmo eficiente para encontrar el camino más largo (con mayor cantidad de aristas) en un árbol.

4. Sea T un árbol generador mínimo producido por el algoritmo de Prim. Pruebe que T contiene todas las aristas de peso mínimo salvo que estas incluyan un circuito.

5. Dado un árbol pesado, definimos su *cuello de botella* como el peso máximo de sus aristas. Muestre que todo árbol generador mínimo minimiza el cuello de botella sobre los árboles generadores de un grafo dado.

6. Sea $G = (V, E)$ un grafo pesado y T un árbol generador mínimo de G. Supongamos que agregamos la arista $e = (u, v)$ de peso p. Describa un algoritmo eficiente para encontrar un árbol generador mínimo de $G + e$.

7. Decida la veracidad de cada una de las siguientes afirmaciones y actúe en consecuencia.
    a) El algoritmo de Dijkstra encuentra caminos mínimos en grafos pesados en que no todos los pesos son no negativos.
    b) El algoritmo de Prim encuentra el árbol generador mínimo en grafos pesados en que no todos los pesos son no negativos.
    c) El único camino entre dos vértices en un árbol generador mínimo de un grafo es un camino mínimo entre ellos.
    d) Supongamos que T es el *único* árbol generador de un grafo. El único camino entre dos vértices de T es un camino mínimo entre ellos.
    e) Sea T un árbol generador mínimo de un grafo pesado G. Se obtiene un nuevo grafo G' a partir de G sumándole k a cada arista de G. Las aristas de T forman un árbol generador mínimo de G.
    f) Sea P un camino de costo mínimo entre s y t en un grafo pesado G. El camino P considerado como camino entre s y t en el grafo modificado G' del ítem anterior es un camino mínimo entre s y t.

8. Sea $G = (V, E)$ un grafo con costos en las aristas. Notemos c_{ij} el costo de la arista que une los vértices v_i y v_j.

   a) Dado π un vector de n coordenadas definimos nuevos costos $c^{\pi}_{ij} := c_{ij} + \pi_j - \pi_i$. Muestre que los caminos mínimos con estos nuevos costos son los mismos que antes (el costo total de cada camino puede cambiar).

   b) Supongamos que G no tiene ciclos negativos y sea s un nodo distinguido en G, tomemos π_i como la longitud del camino mínimo (con los costos originales) desde s hasta v_i. Demuestre que $\pi_j \leq \pi_i + c_{ij}$.

   c) Pruebe que si vale que $\pi_j \leq \pi_i + c_{ij}$ entonces los nuevos costos son no-negativos.

   d) Supongamos que los vértices de G son puntos del plano y los costos son las distancias euclídeas. Demuestre que tomando $\pi_i = \|s - v_i\|^2$ se tiene $\pi_j \leq \pi_i + c_{ij}$.

9. Escriba una modificación del algoritmo de Dijkstra para:

   a) calcular el costo de un camino mínimo entre dos vértices dados de un grafo si las aristas y los vértices tienen pesos.

   b) calcular el costo de un camino entre dos vértices dados que minimice el máximo de los pesos de las aristas que lo componen.

10. Sea $G = (V, E)$ un grafo pesado con pesos no negativos y $s, t \in V$ dos vértices distintos. Decimos que una arista e es $\{s, t\}$-óptima si existe algún camino mínimo entre s y t que pasa por e. Describa un algoritmo para calcular el camino de mínimo costo de s a t que no use arista $\{s, t\}$-óptima alguna.

11. Sea $G = (V, E)$ un grafo dirigido. Diseñe un algoritmo eficiente para calcular el mínimo peso de un ciclo dirigido en G.

12. Sea $G = (V, E)$ un digrafo acíclico pesado (con pesos no necesariamente positivos). Describa un algoritmo eficiente para calcular el camino mínimo entre dos vértices dados.

13. Sea $G = (V, E)$ un grafo dirigido pesado. Sean $u, v \in V$ y $k \leq |V|$ un número natural. Diseñe un algoritmo para encontrar un camino de costo mínimo entre u y v que use exactamente k aristas (no necesariamente distintas).

14. ¿Cómo se puede resolver el problema de flujo máximo en una red si los vértices también tienen capacidades máximas?

15. Sea $G = (V, E)$ un grafo y sean $u, v \in V$. ¿Cuántos caminos disjuntos en aristas se pueden encontrar simultáneamente entre u y v? ¿Y disjuntos en vértices?

16. Sea $a_1, a_2, ..., a_{2n}$ una lista de números naturales sin repeticiones. Se quieren armar parejas con estos números de forma que en cada pareja la suma de ambos números sea primo. ¿Cuál es la máxima cantidad de parejas que se pueden armar? Modele este problema como un problema de flujo. Ejemplo: Si los números son 2,3,4,5 se pueden armar 2 parejas 2-5, 3-4. Pero si son 2,3,4,6, sólo se puede armar una pareja.

17. Dadas g^e y g^s dos n-tuplas de naturales, se desea saber si existe un grafo dirigido G tal que los grados de entrada y salida de cada vértice v_i sean g^e_i y g^s_i respectivamente. Modele como un problema de flujo en redes.

18. En la bolsa de Buenos Aires se hicieron mediciones de los precios de l acciones durante k días. Llamamos $a_{i,j}$ al precio de la acción i en el día j. Para presentar los resultados del estudio vamos a usar gráficos del valor de cada acción en función del tiempo formando una poligonal. Es posible incluir los datos de más de una acción en el mismo gráfico siempre y cuando las poligonales no se corten. Diseñe un algoritmo que calcule el mínimo número de gráficos en el que se pueden acomodar la totalidad de las acciones. *Sugerencia*: Considere el grafo G con vértices $V = \{1, ..., l\}$ y aristas $E = \{(i, i') : \text{si la acción } i \text{ se puede dibujar por abajo de la acción } i'\}$.

19. Encuentre un emparejamiento máximo en el grafo bipartito $G = (V, E)$ siguiente:

    - $V = \{n \in \mathbb{N} : n \leq 13\}$.
    - $E = \{(1, 4), (1, 8), (1, 9), (2, 5), (2, 9), (2, 12), (3, 5), (3, 11), (3, 12), (4, 6), (7, 12), (7, 13), (10, 12)\}$.

20. Muestre cómo resolver el problema de matching máximo bipartito, considerándolo como un problema de flujo máximo en una red conveniente.