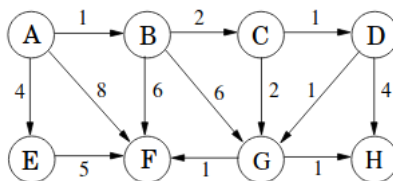


Ejercicios en clase: Caminos minimos en grafos

Análisis y Diseño de Algoritmos

28 de noviembre de 2022

Ejercicio 1. Muestre el valor de $dist$ en cada iteración del algoritmo de Dijkstra, a partir del nodo A , y suponiendo que los vecinos de un vértice son procesados en orden alfabético



Ejercicio 2. Suponga que cambiamos la línea 5 del algoritmo de Dijkstra (implementación con fila de prioridades) como sigue:

while $|Q| > 1$

Eso causa que la condición se ejecute $|V(G)| - 1$ veces en lugar de $|V(G)|$ veces. ¿El algoritmo continúa correcto? Justifique.

Ejercicio 3. Diseñe (haga un pseudocódigo) de un algoritmo que hace lo siguiente:

Recibe: Un grafo no dirigido G con longitudes no negativas ℓ en las aristas y una arista $e \in E(G)$. Devuelve: La longitud de un circuito (ciclo sin repeticiones de vértices ni aristas) mínimo que contiene e .

Su algoritmo deberá consumir tiempo $O(|V(G)|^2)$.

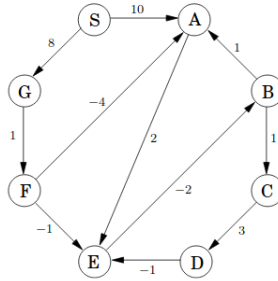
Ejercicio 4. Demuestre la siguiente propiedad. Sea G un grafo dirigido con longitudes ℓ en las aristas y tres vértices u, v, w . Entonces

$$dist(u, v) + dist(v, w) \geq dist(u, w)$$

donde $dist(x, y)$ guarda la distancia de x hacia y .

Ejercicio 5. Muestre el valor de $dist$ en cada iteración del algoritmo Bellman-Ford, desde el nodo A , suponiendo que se revisan las aristas en el siguiente orden:

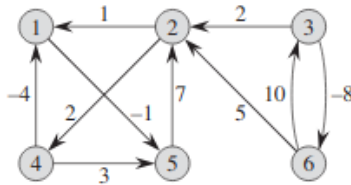
$AE, BA, BC, CD, DE, EB, FA, FE, GF, SA, SG$



Ejercicio 6. Podemos modificar el algoritmo de Bellman-Ford para que termine en menos iteraciones si detecta que no han habido modificaciones en las distancias.

- (a) Diseñe (haga pseudocódigo) que incluya esa modificación
- (b) ¿El número de iteraciones, luego de incluir la modificación sugerida en (a), podría cambiar para una misma entrada si es que el orden de revisión de las aristas cambia?

Ejercicio 7. Ejecute los algoritmos vistos en clase para calcular caminos mínimos entre todos los pares en el siguiente grafo dirigido. Muestre los resultados parciales (iteración por iteración).



Ejercicio 8. Indique como modificar los algoritmos vistos en clase para calcular caminos mínimos entre todos los pares, para que también devuelvan una matriz de predecesores. En esta matriz, la posición (i, j) guarda el predecesor de j en un camino mínimo de i hacia j .