Programación 3
Año 2022

Práctico 3

Ejercicio 1 (Kleinberg & Tardos, Ex. 3.2)**.** Dé un algoritmo para detectar si un grafo G contiene un ciclo. Si la respuesta es afirmativa, el algoritmo debe mostrar un ciclo como evidencia (no todos los ciclos que existan sino solo uno de ellos). Su algoritmo debe admitir una implementación cuyo tiempo de ejecución sea $O(m+n)$, donde n es la cantidad de vértices de G y m es la cantidad de aristas de G.

(a) Presente un algoritmo basado en DFS que resuelve el problema; reescriba cualquier algoritmo que utilice del libro de referencia.

(b) Demuestre la corrección de su algoritmo, esto es, que su algoritmo responde afirmativamente si y solo si G contiene un ciclo y, en dicho caso, que la evidencia mostrada es en efecto un ciclo de G.
Sugerencia:
- Demuestre que si su algoritmo responde afirmativamente, genera como salida una secuencia de vértices (la evidencia pedida) que es un ciclo de G.
- Demuestre que si G contiene un ciclo, su algoritmo responde afirmativamente.

(c) Demuestre que su algoritmo admite una implementación cuyo tiempo de ejecución es $O(m+n)$.
Sugerencia: Realice un análisis similar al de algoritmos para recorrida de grafos.

(d) Explique si es posible resolver el problema sin basarse en DFS.

Ejercicio 2 (Kleinberg & Tardos, Ex. 3.10)**.** En un determinado grafo cada vértice representa una persona vinculada a un escándalo político y las aristas representan algún tipo de relación entre ellas. Un analista está interesado en investigar detenidamente la conexión entre pares de personas u, v que están conectadas por caminos cortos en este grafo. Está claro que la conexión entre algunos de estos pares puede ser puramente casual, por lo cual, además de estudiar el largo del camino más corto entre u, v, decide estudiar la *cantidad* de caminos de largo mínimo entre u, v.

Consideramos un grafo G y un par de vértices u, v de G.

(a) Escriba un algoritmo que calcula la cantidad de caminos de largo mínimo entre u y v (no los caminos en sí mismos, solo la cantidad). El algoritmo debe admitir una implementación cuyo tiempo de ejecución sea $O(m+n)$, donde n es la cantidad de vértices de G y m es la cantidad de aristas de G.

Sugerencia: Diseñe un algoritmo que para cada vértice w calcule la cantidad de caminos de largo mínimo, N_w, desde el vértice u hasta el vértice w.

(b) Demuestre la corrección de su algoritmo.

Sugerencia: Escriba una recurrencia para N_w, definiendo un paso base apropiado y expresando N_w, para $w \neq u$, en función de los valores N_t para vértices t que ocurren inmediatamente antes que w en un camino de largo mínimo desde u hasta w. Demuestre la corrección de esta recurrencia y muestre que su algoritmo la calcula correctamente.

(c) Demuestre que su algoritmo admite una implementación cuyo tiempo de ejecución es $O(m + n)$.

Ejercicio 3 (Kleinberg & Tardos, Ex. 3.4)**.** Un coleccionista de mariposas regresa de una expedición con n especímenes que se sabe que son de dos especies diferentes. Como es muy difícil decidir directamente a qué especie pertenece una mariposa mirándola individualmente, el coleccionista analiza cada par de mariposas con el objetivo de juzgar si son de la misma especie o no. Si está convencido, establece un juicio y etiqueta el par como *iguales* o como *diferentes*; si tiene dudas no establece juicio y no etiqueta el par. Ahora quiere saber si el conjunto de m juicios que estableció es consistente, o sea si es posible asignar a cada mariposa una de dos especies, A o B, de manera que para todo par etiquetado como *iguales* se cumple que las mariposas fueron asignadas a la misma especie y para todo par etiquetado como *diferentes* se cumple que las mariposas fueron asignadas a especies distintas.

(a) Si cada vez que estableció un juicio el par fue etiquetado como *iguales* ¿es posible que el conjunto de juicios sea inconsistente? Si no es posible explique por qué; si es posible escriba un algoritmo que resuelva el problema.

(b) Si cada vez que estableció un juicio el par fue etiquetado como *diferentes* ¿es posible que el conjunto de juicios sea inconsistente? Si no es posible explique por qué; si es posible escriba un algoritmo que resuelva el problema.

(c) Para el caso general (puede haber pares etiquetados como *iguales* y pares etiquetados como *diferentes*) escriba un algoritmo que resuelve el problema.

Sugerencia: Diseñe un algoritmo que asigne una de dos especies a cada mariposa. Demuestre que si esa asignación es inconsistente cualquier otra también lo sería.

Todos los algoritmos deben admitir una implementación cuyo tiempo de ejecución sea $O(m+n)$. Reescriba cualquier algoritmo que utilice el libro de referencia.

(d) Demuestre la corrección de sus algoritmos.

(e) Demuestre que sus algoritmos admiten una implementación cuyo tiempo de ejecución es $O(m+n)$.