Programación 3
Año 2023

Práctico 4

Ejercicio 1 (Kleinberg & Tardos, Ex. 3.9)**.** Una red de comunicaciones consta de n equipos y m enlaces de conexión punto a punto entre ellos. Modelamos esta red como un grafo conexo $G = (V, E)$, donde V es el conjunto de equipos de la red y E contiene m aristas, cada una representando un enlace. En dicha red existe dos equipos, s y t, que están a más de $n/2$ enlaces de distancia. En términos del grafo G, el camino más corto entre los nodos s y t tiene un largo estrictamente mayor a $n/2$ (o sea, el largo es mayor o igual a $\lfloor \frac{n}{2} \rfloor + 1$).

(a) Demuestre que existe un equipo que si se retira de la red deja desconectados a s y t. En otras palabras, existe un nodo que si es eliminado de G deja a los nodos s y t en componentes conexas diferentes.
Sugerencia: Sea L_i, $0 \leq i < n$, el conjunto de nodos de G que están a distancia i de s. Muestre que existe al menos un valor de i, $1 \leq i \leq \left\lfloor \frac{n}{2} \right\rfloor$, tal que L_i contiene un solo nodo.

(b) Escriba un algoritmo que encuentra dicho nodo; debe admitir una implementación cuyo tiempo de ejecución sea $O(m + n)$.

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

Ejercicio 2 (Kleinberg & Tardos, Ex. 3.3)**.** Dado un grafo dirigido arbitrario G (no necesariamente un DAG), se quiere obtener alguno de los siguientes resultados:

- un orden topológico para G,
- un ciclo de G.

(a) Presente un algoritmo que resuelve el problema; reescriba cualquier algoritmo que utilice del libro de referencia.
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.
Sugerencia: Utilice las ideas que se presentan en la demostración del resultado (3.19) del libro de referencia.

(b) Demuestre la corrección de su algoritmo.

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

Ejercicio 3 (Kleinberg & Tardos, Ex. 3.12)**.** A partir de una serie de entrevistas en una investigación histórica se recopilaron una serie de datos sobre un conjunto de n personas, $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$, que ya no están vivas. Los datos recopilados se refieren a pares de personas, (P_i, P_j), en cierto subconjunto de $\mathcal{P} \times \mathcal{P}$, y cada dato puede ser de alguno de los tipos:

1. P_i murió antes de que P_j naciera,
2. las vidas de P_i y P_j se solaparon en el tiempo.

Queremos determinar si nuestros datos son consistentes, es decir, si puede haber existido un conjunto de n personas tales que todos los datos recopilados se satisfacen, respetando además el orden natural de nacimiento y muerte para cada persona. Para ello necesitamos un algoritmo cuya salida sea alguno de los siguientes resultados:

- Una cronología de nacimientos y muertes de las n personas que verifique la consistencia de los datos recopilados.
  Por ejemplo, si para $n = 3$ los datos recopilados indican que P_1 murió antes de que P_3 naciera y que las vidas de P_2 y P_1 se solaparon, la salida de nuestro algoritmo podría ser:

  nace P_2, nace P_1, muere P_1, nace P_3, muere P_2, muere P_3.

- Un mensaje indicando que tal cronología no existe y por lo tanto nuestros datos son inconsistentes.

(a) Presente un algoritmo que resuelve el problema. Describa la representación con que se recibe la entrada (en un formato natural para el problema descrito) y formalice un modelo para resolver el problema. Su algoritmo debe admitir una implementación cuyo tiempo de ejecución sea $O(m + n)$, donde m es la cantidad de datos que fueron recopilados.

(b) Demuestre la corrección de su algoritmo.

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