Programación 3
Año 2023

Práctico 5

Ejercicio 1 (Kleinberg & Tardos, Ex. 4.5)**.** Una compañía de telecomunicaciones necesita desplegar estaciones de acceso a lo largo de una carretera poco poblada para dar cobertura a las casas ubicadas sobre ella. Para garantizar la cobertura se necesita que ninguna casa esté a más de 4 kilómetros de distancia de una estación (para simplificar el problema suponemos que la carretera es recta). Disponemos de la ubicación de cada una de las casas, representada mediante la distancia lineal contando desde el inicio de la carretera.

(a) Dé un algoritmo eficiente para determinar la ubicación de las estaciones de manera que se satisfaga el requerimiento de cobertura con la menor cantidad de estaciones posible.
(b) Demuestre la corrección de su algoritmo.
Sugerencia: Siga una estrategia similar a la empleada en la sección 4.1 del libro.

Ejercicio 2 (Kleinberg & Tardos, Ex. 4.6)**.** La organización de un triatlón amateur cuenta, para cada participante inscripto, con una estimación del tiempo que le llevará completar cada una de las etapas: natación, ciclismo y carrera a pie. La piscina donde se desarrolla la primera etapa solo puede ser ocupada por un nadador a la vez; las etapas posteriores pueden desarrollarse simultáneamente por varios competidores. Queremos determinar en qué orden organizar la largada de los competidores (necesariamente uno a la vez) de modo de minimizar la duración estimada de la competencia completa, que queda determinada cuando el último participante llega a la meta de la última etapa.
(a) Dé un algoritmo eficiente para resolver el problema.
(b) Demuestre la corrección de su algoritmo.

Ejercicio 3 (Kleinberg & Tardos, Ex. 4.18)**.** Estamos planificando un viaje desde un *origen* o a un *destino* d. Luego de investigar un poco se obtiene un *mapa* que además de o y d incluye todos los posibles puntos intermedios por los que se podría pasar para llegar al destino. Todos los puntos del mapa son alcanzables desde el origen. Un punto u está *conectado directamente* a otro v si se puede viajar desde u hacia v sin pasar por puntos intermedios. Que u esté conectado directamente a v no implica necesariamente que v esté conectado directamente a u. Contamos también con un servicio web para estimar el tiempo requerido para viajar en determinado momento desde un punto a otro al que está conectados directamente. Específicamente, dados dos puntos, u y v, y un tiempo t en el que esperamos comenzar a viajar desde u hacia v, el servicio es capaz de calcular y devolvernos un *tiempo estimado de llegada* a v, que representamos mediante una función $f_{uv}(t)$. Hay que notar que el *tiempo de viaje* es $f_{uv}(t) - t$, y no es constante, ya que puede depender de condiciones de tráfico, condiciones climáticas, etc. Además, el servicio web garantiza:

- $f_{uv}(t) \geq t, \forall t$ (no se puede ir hacia atrás en el tiempo);
- $f_{uv}(t_1) \leq f_{uv}(t_2), \forall t_1 \leq t_2$ (f_{uv} es no decreciente con t implicando que no se llega antes a v si se comienza a viajar desde u más tarde).

Nos interesa encontrar la forma más rápida de llegar al destino comenzando desde el origen en tiempo $t = 0$.

(a) Escriba un algoritmo que resuelve el problema a partir de una lista de m pares de puntos directamente conectados (cada punto tiene un identificador que pertenece a $[1 \,..\, n]$), donde n es la cantidad de puntos del mapa y m es la cantidad de pares de puntos conectados directamente. El algoritmo debe admitir una implementación cuyo tiempo de ejecución es $O(m \log n)$. El tiempo requerido para el cálculo de $f_{uv}(t)$ es $O(1), \forall u, v, t$.

(b) Demuestre la corrección de su algoritmo.
Sugerencia: Siga la misma línea de razonamiento usada en el análisis del algoritmo de Dijkstra, teniendo en cuenta las hipótesis sobre las funciones f_{uv}.

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