



Ayudantía 12 de Junio

IIC2133 - Estructuras de datos y algoritmos

Primer semestre, 2017

Problema 1

Para un grafo $G(V, E)$ se quiere calcular el árbol de cobertura mínima, teniendo en cuenta que en el futuro se querrán insertar nuevas aristas o eliminar aristas existentes en E . Proponga un algoritmo que permita calcular el árbol, de manera de que además le permita recomputar, eficientemente, el árbol frente a estos cambios.

Se espera que la complejidad de volver a calcular el árbol frente a un cambio sea $O(|E|)$

Problema 2

Para un conjunto de tareas $T = \{ t_1, \dots, t_n \}$ se tiene una lista de requisitos $R = \{ \dots, (t_i, t_j), \dots \}$ que contiene la relación (t_i, t_j) si y solo si la tarea t_i debe necesariamente ser realizada antes que la tarea t_j . Adicionalmente, se tiene una lista de costos $C = \{ c_1, \dots, c_n \}$ para las tareas, donde el costo de la tarea t_i es c_i . Se desea obtener una lista de las tareas en el orden secuencial en que deberían ser realizadas, considerando que se prefiere siempre llevar a cabo tareas más baratas primero.

Considere usar una modificación del algoritmo de orden topológico sobre el grafo $G(T, R)$, para obtener una complejidad cercana a $O(|V| + |E|)$

Problema 3

Considere que se desea hacer una aplicación online de compra y venta instantánea al mejor postor, y se desean implementar los siguientes comandos para los usuarios:

- **COMPRAR**(u, x, c, pc): el usuario u comunica que quiere comprar c unidades del producto x a un precio pc . En caso de que el producto esté a la venta, y su precio de venta sea menor a pc , entonces el cliente efectivamente comprará el producto, asumiendo que no hay otro usuario esperando. De no haber stock del producto, el cliente quedará en espera para cuando vuelvan a haber unidades del producto a la venta. Si el cliente ya está en espera para este producto al usar este comando, entonces deberá actualizarse su precio de compra y cantidad de unidades.
- **VENDER**(u, x, c, pv): el usuario u pone a la venta c unidades del producto x al precio p , vendiéndolo instantáneamente a las c personas que más hayan ofrecido por el producto, siempre que hayan ofrecido más que pv . En caso de que no se vendan todas las unidades, quedarán en stock, vendiéndose al cliente que más ofrezca, mientras su precio de compra sea mayor o igual al precio de venta pv . Si ya hay unidades de ese producto a la venta, entonces las cantidades en stock se suman, pero los precios se manejan de manera individual para el producto. Siempre se preferirá vender a un usuario la versión más barata de un mismo producto.

Explique como debe hacer este sistema para poder responder a estas consultas de manera eficiente, y que estructuras de datos debe utilizar para ello.