Parcial de Programación 3
28 de setiembre de 2017

- Este parcial dura **3** horas y media y consta de 2 carillas. El total de puntos es **50**.
- **NO** se puede utilizar ningún tipo de material de consulta. Salvo que se indique lo contrario podrá usarse todo lo visto en el teórico, práctico y laboratorio sin demostrarlo, indicando claramente lo que se está usando.

Se requiere:

I) Numerar todas las hojas e incluir en cada una el **nombre, cédula de identidad, número de página y cantidad de hojas entregadas**.

II) Utilizar las hojas de **un solo lado** y escribir con lápiz.

III) Iniciar cada ejercicio en hoja nueva.

Ejercicio 1 (20 puntos)

Roque es un típico personaje principal de telenovela, y por ende sufrió un ataque de amnesia. Lamentablemente, Roque ya no sabe en qué orden ponerse las prendas a la hora de vestirse, y se ha encontrado múltiples veces con los zapatos puestos antes que las medias. Roque está cansado de que esto lo haga llegar tarde al trabajo y decide contratarte a ti para que le soluciones su problema. Asuma que Roque debe ponerse todas las prendas que tiene en su ropero.

(a) Modele la relación de precedencia entre las prendas de vestir mediante un grafo (esto es, que para que Roque se pueda poner un cinturón debe tener puesto antes un pantalón). ¿Qué características tiene el grafo propuesto?

(b) Describa en pseudocódigo un algoritmo que le permita a Roque determinar en qué orden ponerse las prendas, de manera que una vez puesta una prenda no deba sacársela. El algoritmo debe recibir como entrada el grafo propuesto de la parte (a) y devolver una secuencia de prendas.

(c) Roque tiene la peculiar habilidad de ponerse en 10 segundos cualquier número de prendas simultáneamente, mientras no existan relaciones de precedencia entre ellas (si es una sola prenda también le lleva 10 segundos). ¿Qué información debería saber del grafo de la parte (a) para poder determinar el tiempo mínimo que le llevaría a Roque vestirse? ¿Cuál sería ese tiempo?

Ejercicio 2 (15 puntos)

(a) Considere el grafo conexo no dirigido $G = (V, E)$, cuyas aristas e tienen asociados costos c_e, todos distintos. Sea C un ciclo cualquiera de G, y $e = (v, w)$ la arista más costosa de C. Demuestre que e no pertenece a ningún árbol de cubrimiento mínimo (*minimum spanning tree*) de G.

(b) Sea $G = (V, E)$ un grafo conexo no dirigido con costos $c_e \geq 0$ sobre las aristas $e \in E$. Asuma que se le ofrece un árbol de cubrimiento mínimo (*minimum-cost spanning tree*) T de G. Ahora asuma que a G se le agrega una arista nueva de costo c que conecta los nodos $v, w \in V$.

Escriba un algoritmo eficiente que verifique si T sigue siendo un árbol de cubrimiento mínimo del nuevo grafo G. Asuma que los costos c_e son todos distintos.

Ejercicio 3 (15 puntos)

(a) Describa formalmente en que consiste el problema de *"El par de puntos más cercanos"*, tratado en el curso.

La solución vista cumple la relación de recurrencia

$$T(n) \leq \begin{cases} 0 & \text{si } n = 1, \\ 2T(n/2) + c(n \log n) & \text{en otro caso.} \end{cases}$$

siendo c una constante mayor que 0. Explique brevemente la procedencia de esos términos.

(b) La mecánica Mechy tiene una caja con n tuercas y sus correspondientes tornillos. Todas las tuercas tienen diámetros diferentes. No se puede comparar un par de tuercas entre sí ni un par de tornillos entre sí. Pero se puede comparar cualquier tuerca con cualquier tornillo y determinar el tamaño relativo del tornillo con respecto a la tuerca. El propósito es ayudar a Mechy a enroscar cada tornillo con su tuerca correspondiente y se pretende lograrlo con $O(n \log n)$ comparaciones entre tornillos y tuercas en el caso promedio.

Describa en lenguaje natural (no pseudocódigo) el algoritmo *divide & conquer* para lograrlo, dejando claro cuáles son los subproblemas en que se divide el problema original. ¿Cuándo se da el peor caso y cuál es el orden?