

1.º recuperatorio de examen parcial – 24/06/2024

1. Juan es ambicioso pero también algo vago. Dispone de varias ofertas de trabajo diarias, pero no quiere trabajar tres días seguidos. Se tiene la información de la ganancia del día  $i$  ( $G_i$ ), para cada día. Implementar un modelo de **programación lineal** que maximice el monto a ganar por Juan, sabiendo que no aceptará trabajar tres días seguidos.
2. Implementar un algoritmo greedy que permita obtener el mínimo del problema del viajante: dado un Grafo pesado  $G$  y un vértice de inicio  $v$ , obtener el camino de menor costo que lleve a un viajante desde  $v$  hacia cada uno de los vértices del grafo, pasando por cada uno de ellos una única vez, y volviendo nuevamente al origen. Se puede asumir que el grafo es completo. Indicar y justificar la complejidad del algoritmo implementado.

¿El algoritmo obtiene siempre la solución óptima? Si es así, justificar detalladamente, sino dar un contraejemplo. Indicar y justificar la complejidad del algoritmo implementado. Justificar por qué se trata de un algoritmo greedy.

3. Coty cumplió años ayer y está organizando su festejo. En dicho festejo, va a dar unos regalos. Son regalos geniales, que van a dar que hablar luego del festejo. Eso es justamente lo que desea ella: que todos aquellos invitados que se conozcan entre sí, luego de terminado el evento hablen del regalo que recibió uno, o bien el otro. ¿El problema? Coty está invitando a  $n$  personas, pero no tiene presupuesto para comprar  $n$  regalos, sino tan sólo  $k$ .

*El problema del cumpleaños de Coty* puede enunciarse como: Dada la lista de  $n$  invitados al cumpleaños de Coty, un número  $k$ , y conociendo quién se conocen con quién (ej: una lista con los pares de conocidos), ¿existe una forma de asignar a lo sumo  $k$  personas para dar los regalos, de tal forma que todos los invitados, al hablar luego con quienes se conozcan, puedan hablar del regalo que obtuvo uno o bien el otro?

Demstrar que *el problema del cumpleaños de Coty* es un problema NP-Completo.

4. Implementar un algoritmo **potencia(b, n)** que nos devuelva el resultado de  $b^n$  en tiempo  $\mathcal{O}(\log n)$ . Justificar adecuadamente la complejidad del algoritmo implementado. *Ayuda:* recordar propiedades matemáticas de la potencia. Por ejemplo, que  $a^h \cdot a^k = a^{h+k}$ .
5. Dado un número  $n$ , mostrar la cantidad más económica (con menos términos) de escribirlo como una suma de cuadrados, utilizando programación dinámica. Indicar y justificar la complejidad del algoritmo implementado (cuidado con esto, es fácil tentarse a dar una cota más alta de lo correcto). Implementar un algoritmo que permita reconstruir la solución.

*Aclaración:* siempre es posible escribir a  $n$  como suma de  $n$  términos de la forma  $1^2$ , por lo que siempre existe solución. Sin embargo, la expresión  $10 = 3^2 + 1^2$  es una manera más económica de escribirlo para  $n = 10$ , pues sólo tiene dos términos.

1.º recuperatorio de examen parcial – 24/06/2024

1. Juan es ambicioso pero también algo vago. Dispone de varias ofertas de trabajo diarias, pero no quiere trabajar tres días seguidos. Se tiene la información de la ganancia del día  $i$  ( $G_i$ ), para cada día. Implementar un modelo de **programación lineal** que maximice el monto a ganar por Juan, sabiendo que no aceptará trabajar tres días seguidos.
2. Implementar un algoritmo greedy que permita obtener el mínimo del problema del viajante: dado un Grafo pesado  $G$  y un vértice de inicio  $v$ , obtener el camino de menor costo que lleve a un viajante desde  $v$  hacia cada uno de los vértices del grafo, pasando por cada uno de ellos una única vez, y volviendo nuevamente al origen. Se puede asumir que el grafo es completo. Indicar y justificar la complejidad del algoritmo implementado.

¿El algoritmo obtiene siempre la solución óptima? Si es así, justificar detalladamente, sino dar un contraejemplo. Indicar y justificar la complejidad del algoritmo implementado. Justificar por qué se trata de un algoritmo greedy.

3. Coty cumplió años ayer y está organizando su festejo. En dicho festejo, va a dar unos regalos. Son regalos geniales, que van a dar que hablar luego del festejo. Eso es justamente lo que desea ella: que todos aquellos invitados que se conozcan entre sí, luego de terminado el evento hablen del regalo que recibió uno, o bien el otro. ¿El problema? Coty está invitando a  $n$  personas, pero no tiene presupuesto para comprar  $n$  regalos, sino tan sólo  $k$ .

*El problema del cumpleaños de Coty* puede enunciarse como: Dada la lista de  $n$  invitados al cumpleaños de Coty, un número  $k$ , y conociendo quién se conocen con quién (ej: una lista con los pares de conocidos), ¿existe una forma de asignar a lo sumo  $k$  personas para dar los regalos, de tal forma que todos los invitados, al hablar luego con quienes se conozcan, puedan hablar del regalo que obtuvo uno o bien el otro?

Demstrar que *el problema del cumpleaños de Coty* es un problema NP-Completo.

4. Implementar un algoritmo **potencia(b, n)** que nos devuelva el resultado de  $b^n$  en tiempo  $\mathcal{O}(\log n)$ . Justificar adecuadamente la complejidad del algoritmo implementado. *Ayuda:* recordar propiedades matemáticas de la potencia. Por ejemplo, que  $a^h \cdot a^k = a^{h+k}$ .
5. Dado un número  $n$ , mostrar la cantidad más económica (con menos términos) de escribirlo como una suma de cuadrados, utilizando programación dinámica. Indicar y justificar la complejidad del algoritmo implementado (cuidado con esto, es fácil tentarse a dar una cota más alta de lo correcto). Implementar un algoritmo que permita reconstruir la solución.

*Aclaración:* siempre es posible escribir a  $n$  como suma de  $n$  términos de la forma  $1^2$ , por lo que siempre existe solución. Sin embargo, la expresión  $10 = 3^2 + 1^2$  es una manera más económica de escribirlo para  $n = 10$ , pues sólo tiene dos términos.