

Examen parcial (1R) – 05/12/2024

1. Implementar un algoritmo que dado un grafo **no dirigido** y conexo, un vértice v y otro w , determine la cantidad máxima de caminos disjuntos, y cómo son, que conectan a v con w . Indicar y justificar la complejidad del algoritmo implementado.
2. Tati empezó a trabajar en un laboratorio químico, donde trabaja con compuestos que se vaporizan o subliman muy rápidamente, por lo que suelen estar almacenados en congeladores. ¿El problema? Su predecesora, Nerea, tenía un único trabajo y lo hizo mal: en vez de almacenar todos los compuestos en un único congelador, guardó cada compuesto en un congelador por separado.

La empresa no puede sostener un costo tan elevado, y deben deshacerse de todos los congeladores (salvo 1). Esto se resuelve tan simple como pasar todos los compuestos a un único congelador, pero desde el momento que uno de estos se abre, se empieza a perder contenido de los compuestos que lleva dentro porque se vuelven gas, y desean minimizar las pérdidas que esto produzca. Lo que se puede hacer es, en una unidad de tiempo, abrir dos congeladores A y B, y mover todo lo que haya en el congelador A al B. Se perderá en el medio el equivalente de lo que pierde cada componente por unidad de tiempo (dato conocido para cada uno). Se puede seleccionar cualquier par de congeladores para hacer lo antes mencionado. Ejemplo: si tengo el congelador A con el compuesto c_1 que pierde 5 por unidad de tiempo, y el congelador B con el compuesto c_2 que pierde 3 por unidad de tiempo, mover lo del congelado A al B nos implica una pérdida de $5 + 3 = 8$. Si ahora movemos lo que hay en el congelador B al congelador C (que tiene los componentes c_3 , c_4 y c_5 con pérdidas de 7, 4 y 1 respectivamente), el costo de ese movimiento será $5 + 3 + 7 + 4 + 1 = 20$, lo cual se suma, obviamente, a cualquier otra pérdida anterior antes incurrida.

Implementar un algoritmo **greedy** que obtenga el mínimo de pérdida que se puede lograr para terminar con un único congelador. A fines del parcial no es necesario indicar cómo es este proceso, sólo el valor final. Indicar y justificar la complejidad del algoritmo. Justificar por qué el algoritmo implementado es, en efecto, un algoritmo greedy. ¿Es el algoritmo implementado óptimo? Si es, dar una breve explicación, si no lo es dar un contraejemplo.

3. Antonina está a punto de casarse y tiene un problema: no sabe cómo armar las mesas de invitados de la fiesta, de tal manera que todos pasen un buen rato. Desea armar mesas donde todos se lleven bien con todos. Empieza a armar e intentar, pero siempre llega a algún problema, considerando que sólo cuenta con K mesas. Antonina le cuenta de la situación a su futuro esposo, quien tal vez podría ser útil para resolver el problema... Pero no. Él decide explicarle que, en efecto, este es un problema muy difícil. Antonina, resignada a que se está casando con un nerd, decide preguntar algo de lo que luego se arrepentiría: “¿en serio? ¿por qué?” a lo que, muy alegremente, su futuro esposo le responde: “Porque este problema es NP-Completo, mirá, te lo demuestro...”.

El problema del casamiento de Antonina puede enunciarse como: Dado un conjunto de invitados, un listado donde en cada entrada tiene 2 personas que se llevan bien entre sí, y un número K , ¿existe una forma de armar hasta K mesas para la fiesta del casamiento de tal forma que todos los integrantes de una mesa se lleven bien?

Demostrar que el problema del casamiento de Antonina es un problema NP-Completo. Tal vez quieras recordar que, después del primer examen, sabemos que el problema de *Separación en R cliques* es un problema NP-Completo. Sino, otro problema que podés utilizar es el de coloreo.

4. Implementar un algoritmo que, por **programación dinámica**, resuelva el problema de la mochila con una variante: ahora se puede poner una cantidad ilimitada de un mismo elemento (es decir, se puede repetir), siempre y cuando aún haya lugar. Por ejemplo, si yo tengo un elemento de tamaño 3 y una mochila de tamaño 10, yo podría guardar 3 veces dicho elemento, si así lo quisiera (también menos cantidad). Escribir y describir la ecuación de recurrencia de la solución, y la complejidad del algoritmo implementado. Implementar o explicar (la que prefieran) cómo sería el algoritmo de reconstrucción de la solución, indicando su complejidad.
5. Osvaldo es un empleado de una inescrupulosa empresa inmobiliaria, y está buscando un ascenso. Está viendo cómo se predice que evolucionará el precio de un inmueble (el cual no poseen, pero pueden comprar). Tiene la información de estas predicciones en el arreglo p , para todo día $i = 1, 2, \dots, n$. Osvaldo quiere determinar un día j en el cuál comprar la casa, y un día k en el cual venderla ($k > j$), suponiendo que eso sucederá sin lugar a dudas. El objetivo, por supuesto, es el de maximizar la ganancia dada por $p[k] - p[j]$.

Implementar un modelo de **programación lineal** que determine el día de compra y el día de venta del inmueble. Indicar la cantidad de restricciones implementadas para esto.