

Examen parcial – 16/02/2024

1. Se define el problema *2-Partition* de la siguiente manera: Se cuenta con un conjunto de n elementos. Cada uno de ellos tiene un valor asociado. Se desea separar los elementos en 2 subconjuntos tal que la suma de los valores de cada subconjunto sea igual para ambos.
2-Partition es un problema NP-Completo. Queremos demostrar nuevamente (pero de otra forma a la vista en clase) que *Subset Sum* es un problema NP-Completo. Demostrar esto, utilizando que *2-Partition* es un problema NP-Completo.
2. En clase vimos una solución óptima del problema del cambio utilizando programación dinámica. Ahora planteamos un problema similar: Implementar un algoritmo que dado un set de monedas posibles y una cantidad de cambio a dar, devuelva **la cantidad de formas diferentes** que hay para dar dicho cambio. El algoritmo a implementar debe ser también por programación dinámica. Indicar y justificar la complejidad del algoritmo implementado. Importante: antes de escribir código, escribir (y describir) la ecuación de recurrencia.
3. Implementar un algoritmo que reciba un grafo no dirigido y un número k , y devuelva un ciclo de tamaño exactamente k del grafo, si es que existe.
4. La Nación del Fuego está atacando al Reino de la Tierra en su capital, Ba Sing Se. Es conocido el eficiente sistema de riego de la ciudad, que le permite estar encerrada sin problemas. A la ciudad entran k ríos, y mediante un sistema de tuberías logran encausar el agua hacia los m puntos de riego. Gracias a un espía, la Nación del Fuego obtuvo un plano completo de la red, incluyendo cuánta agua puede transportar cada tubería, etc... El Señor del Fuego ha ideado un plan: infiltrar a un atacante que destruya un único punto de conexión de tuberías (no creen que poder tener más tiempo que para esto). La conexión debería ser la que reduzca lo máximo posible la cantidad de agua que llega a las zonas de riego. Implementar un algoritmo que determine cuál conexión de la red debería atacar la Nación del Fuego. Indicar y justificar la complejidad del algoritmo implementado.
5. El famoso ladrón Francesco Rizzoli (hermano del “árbitro” de la final del 2014), ha decidido hacer un atraco a un laboratorio farmacéutico. Allí puede robarse diferentes fármacos que se están estudiando (en formato líquido). Tiene un catálogo del valor de cada fármaco, que puede vender en el mercado negro. De cada fármaco hay una diferente cantidad disponible (medible en ml). Rizzoli sólo tiene posibilidad en su equipo de llevarse como máximo L ml en fármacos. Lo bueno es que sabe que puede fraccionar y poner proporciones de los fármacos; y en ese caso lo vendería en su valor proporcional. Implementar un algoritmo greedy que obtenga los fármacos (y cantidades) que Rizzoli debe robarse para obtener la máxima ganancia posible (el algoritmo debe ser óptimo, en esta familia no se aceptan los robos a medias). Justificar por qué el algoritmo propuesto es Greedy. Indicar y justificar la complejidad del algoritmo implementado.

Examen parcial – 16/02/2024

1. Se define el problema *2-Partition* de la siguiente manera: Se cuenta con un conjunto de n elementos. Cada uno de ellos tiene un valor asociado. Se desea separar los elementos en 2 subconjuntos tal que la suma de los valores de cada subconjunto sea igual para ambos.
2-Partition es un problema NP-Completo. Queremos demostrar nuevamente (pero de otra forma a la vista en clase) que *Subset Sum* es un problema NP-Completo. Demostrar esto, utilizando que *2-Partition* es un problema NP-Completo.
2. En clase vimos una solución óptima del problema del cambio utilizando programación dinámica. Ahora planteamos un problema similar: Implementar un algoritmo que dado un set de monedas posibles y una cantidad de cambio a dar, devuelva **la cantidad de formas diferentes** que hay para dar dicho cambio. El algoritmo a implementar debe ser también por programación dinámica. Indicar y justificar la complejidad del algoritmo implementado. Importante: antes de escribir código, escribir (y describir) la ecuación de recurrencia.
3. Implementar un algoritmo que reciba un grafo no dirigido y un número k , y devuelva un ciclo de tamaño exactamente k del grafo, si es que existe.
4. La Nación del Fuego está atacando al Reino de la Tierra en su capital, Ba Sing Se. Es conocido el eficiente sistema de riego de la ciudad, que le permite estar encerrada sin problemas. A la ciudad entran k ríos, y mediante un sistema de tuberías logran encausar el agua hacia los m puntos de riego. Gracias a un espía, la Nación del Fuego obtuvo un plano completo de la red, incluyendo cuánta agua puede transportar cada tubería, etc... El Señor del Fuego ha ideado un plan: infiltrar a un atacante que destruya un único punto de conexión de tuberías (no creen que poder tener más tiempo que para esto). La conexión debería ser la que reduzca lo máximo posible la cantidad de agua que llega a las zonas de riego. Implementar un algoritmo que determine cuál conexión de la red debería atacar la Nación del Fuego. Indicar y justificar la complejidad del algoritmo implementado.
5. El famoso ladrón Francesco Rizzoli (hermano del “árbitro” de la final del 2014), ha decidido hacer un atraco a un laboratorio farmacéutico. Allí puede robarse diferentes fármacos que se están estudiando (en formato líquido). Tiene un catálogo del valor de cada fármaco, que puede vender en el mercado negro. De cada fármaco hay una diferente cantidad disponible (medible en ml). Rizzoli sólo tiene posibilidad en su equipo de llevarse como máximo L ml en fármacos. Lo bueno es que sabe que puede fraccionar y poner proporciones de los fármacos; y en ese caso lo vendería en su valor proporcional. Implementar un algoritmo greedy que obtenga los fármacos (y cantidades) que Rizzoli debe robarse para obtener la máxima ganancia posible (el algoritmo debe ser óptimo, en esta familia no se aceptan los robos a medias). Justificar por qué el algoritmo propuesto es Greedy. Indicar y justificar la complejidad del algoritmo implementado.