Teoría de Algoritmos (TB024) – Curso Buchwald - Genender

Examen parcial – 04/11/2024

1. Resolver, utilizando **backtracking**, el problema de la mochila con cantidades mínimas. Este tiene el mismo planteo al original pero además cuenta con un parámetro K, donde además de las condiciones impuestas para el problema original, se deben utilizar al menos K elementos. Es decir, el planteo completo es: Dados n elementos de valores $v_1, v_2, ..., v_n$ con pesos $p_1, p_2, ..., p_n$, y valores W y K, encontrar el subconjunto de al menos K elementos, cuya suma de valor sea máxima y cuyo peso no exceda el valor de W.

2. El problema de *Separación en R Cliques* (SRC) se enuncia como: Dado un grafo, y un valor entero R, ¿se pueden separar todos los vértices del gráfo en a lo sumo R cliques? (cada clique puede tener una cantidad diferente de vértices). De una manera más formal, se puede enunciar: ¿existen $S_1, S_1, ..., S_k$, subconjuntos **disjuntos** del conjunto de vértices V tal que $\bigcup_i S_i = V$, $k \leq R$, y que que cada subgrafo correspondiente a los S_i sea un clique (subgrafo completo)?

   Demostrar que el problema de *Separación en R Cliques* es un problema NP-Completo. Para esto, recomendamos recordar que el problema de coloreo es un problema NP-Completo. También, recomendamos recordar cómo fue que demostramos en clase que K-Clique es un problema NP-Completo (fue con la ayuda de Independent Set).

3. 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 $i = 1, 2, ..., n$. 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 la de maximizar la ganancia dada por $p[k] - p[j]$.

   Implementar un algoritmo de **programación dinámica** que permita resolver el problema de Osvaldo. Indicar y justificar la complejidad del algoritmo implementado.

4. Implementar un modelo de **programación lineal** que permita determinar el clique de tamaño máximo dentro de un grafo. Indicar la cantidad de restricciones generadas en función de la cantidad de vértices y aristas.

5. Resolver el problema de Osvaldo (ejercicio 3) pero por **división y conquista**. Indicar y justificar adecuadamente la complejidad del algoritmo implementado. Es probable que la complejidad de ambas soluciones no quede igual, no te estreses por ello.

Teoría de Algoritmos (TB024) – Curso Buchwald - Genender

Examen parcial – 04/11/2024

1. Resolver, utilizando **backtracking**, el problema de la mochila con cantidades mínimas. Este tiene el mismo planteo al original pero además cuenta con un parámetro K, donde además de las condiciones impuestas para el problema original, se deben utilizar al menos K elementos. Es decir, el planteo completo es: Dados n elementos de valores $v_1, v_2, ..., v_n$ con pesos $p_1, p_2, ..., p_n$, y valores W y K, encontrar el subconjunto de al menos K elementos, cuya suma de valor sea máxima y cuyo peso no exceda el valor de W.

2. El problema de *Separación en R Cliques* (SRC) se enuncia como: Dado un grafo, y un valor entero R, ¿se pueden separar todos los vértices del gráfo en a lo sumo R cliques? (cada clique puede tener una cantidad diferente de vértices). De una manera más formal, se puede enunciar: ¿existen $S_1, S_1, ..., S_k$, subconjuntos **disjuntos** del conjunto de vértices V tal que $\bigcup_i S_i = V$, $k \leq R$, y que que cada subgrafo correspondiente a los S_i sea un clique (subgrafo completo)?

   Demostrar que el problema de *Separación en R Cliques* es un problema NP-Completo. Para esto, recomendamos recordar que el problema de coloreo es un problema NP-Completo. También, recomendamos recordar cómo fue que demostramos en clase que K-Clique es un problema NP-Completo (fue con la ayuda de Independent Set).

3. 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 $i = 1, 2, ..., n$. 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 la de maximizar la ganancia dada por $p[k] - p[j]$.

   Implementar un algoritmo de **programación dinámica** que permita resolver el problema de Osvaldo. Indicar y justificar la complejidad del algoritmo implementado.

4. Implementar un modelo de **programación lineal** que permita determinar el clique de tamaño máximo dentro de un grafo. Indicar la cantidad de restricciones generadas en función de la cantidad de vértices y aristas.

5. Resolver el problema de Osvaldo (ejercicio 3) pero por **división y conquista**. Indicar y justificar adecuadamente la complejidad del algoritmo implementado. Es probable que la complejidad de ambas soluciones no quede igual, no te estreses por ello.