

Teoría de Algoritmos I (75.29, 95.06) – Curso Buchwald - Genender

5.ª fecha de examen final – 11/08/2023

1. Implementar un algoritmo que, por backtracking, obtenga todos los posibles ordenamientos topológicos de un grafo dirigido y acíclico.
2. Dado un Grafo, un Vertex Cover es el mínimo Subset de vértices, tal que se cubran todas las aristas, es decir, que cada arista tenga por lo menos uno de sus extremos perteneciendo al Vertex Cover. El uso de cada Vértice v_i tiene un costo w_i . Queremos minimizar la sumatoria de los pesos de los vértices utilizados.
 - a. Mostrar alguna metodología que nos permita resolver este problema aproximadamente. Demostrar que se trata de una 2-Aproximación.
 - b. ¿Podemos aprovechar el Vertex Cover resultante para resolver aproximadamente el problema de Independent Set? Justificar.
3. Dada una soga de n metros ($n \geq 2$) implementar un algoritmo que, utilizando programación dinámica, permita cortarla (en partes de largo entero) de manera tal que el producto del largo de cada una de las partes resultantes sea máximo. El algoritmo debe devolver el valor del producto máximo alcanzable. Indicar y justificar la complejidad del algoritmo.

Ejemplos:

$n = 2 \rightarrow$ Debe devolver 1 (producto máximo es $1 * 1$)
 $n = 3 \rightarrow$ Debe devolver 2 (producto máximo es $2 * 1$)
 $n = 5 \rightarrow$ Debe devolver 6 (producto máximo es $2 * 3$)
 $n = 10 \rightarrow$ Debe devolver 36 (producto máximo es $3 * 3 * 4$)

4. Se tiene una matriz donde en cada celda hay submarinos, o no, y se quiere poner faros para iluminarlos a todos. Implementar un algoritmo que **Greedy** que dé la cantidad mínima de faros que se necesitan para que todos los submarinos queden iluminados, siendo que cada faro ilumina su celda y además todas las adyacentes (incluyendo las diagonales), y las directamente adyacentes a estas (es decir, un “radio de 2 celdas”). Indicar y justificar la complejidad del algoritmo implementado. ¿El algoritmo implementado da siempre la solución óptima?
5. El problema de elección de caminos (Path Selection) pregunta: dado un grafo *dirigido* G y un set de pedidos P_1, P_2, \dots, P_c de caminos dentro de dicho grafo y un número k , ¿es posible seleccionar al menos k de esos caminos tales que ningún par de ellos compartan ningún nodo?

Demostrar que Path Selection es un problema NP-Completo. Ayuda: este problema tiene mucha semejanza con Independent Set. Recomendamos reducir dicho problema a Path Selection.

Teoría de Algoritmos I (75.29, 95.06) – Curso Buchwald - Genender

5.ª fecha de examen final – 11/08/2023

1. Implementar un algoritmo que, por backtracking, obtenga todos los posibles ordenamientos topológicos de un grafo dirigido y acíclico.
2. Dado un Grafo, un Vertex Cover es el mínimo Subset de vértices, tal que se cubran todas las aristas, es decir, que cada arista tenga por lo menos uno de sus extremos perteneciendo al Vertex Cover. El uso de cada Vértice v_i tiene un costo w_i . Queremos minimizar la sumatoria de los pesos de los vértices utilizados.
 - a. Mostrar alguna metodología que nos permita resolver este problema aproximadamente. Demostrar que se trata de una 2-Aproximación.
 - b. ¿Podemos aprovechar el Vertex Cover resultante para resolver aproximadamente el problema de Independent Set? Justificar.
3. Dada una soga de n metros ($n \geq 2$) implementar un algoritmo que, utilizando programación dinámica, permita cortarla (en partes de largo entero) de manera tal que el producto del largo de cada una de las partes resultantes sea máximo. El algoritmo debe devolver el valor del producto máximo alcanzable. Indicar y justificar la complejidad del algoritmo.

Ejemplos:

$n = 2 \rightarrow$ Debe devolver 1 (producto máximo es $1 * 1$)
 $n = 3 \rightarrow$ Debe devolver 2 (producto máximo es $2 * 1$)
 $n = 5 \rightarrow$ Debe devolver 6 (producto máximo es $2 * 3$)
 $n = 10 \rightarrow$ Debe devolver 36 (producto máximo es $3 * 3 * 4$)

4. Se tiene una matriz donde en cada celda hay submarinos, o no, y se quiere poner faros para iluminarlos a todos. Implementar un algoritmo que **Greedy** que dé la cantidad mínima de faros que se necesitan para que todos los submarinos queden iluminados, siendo que cada faro ilumina su celda y además todas las adyacentes (incluyendo las diagonales), y las directamente adyacentes a estas (es decir, un “radio de 2 celdas”). Indicar y justificar la complejidad del algoritmo implementado. ¿El algoritmo implementado da siempre la solución óptima?
5. El problema de elección de caminos (Path Selection) pregunta: dado un grafo *dirigido* G y un set de pedidos P_1, P_2, \dots, P_c de caminos dentro de dicho grafo y un número k , ¿es posible seleccionar al menos k de esos caminos tales que ningún par de ellos compartan ningún nodo?

Demostrar que Path Selection es un problema NP-Completo. Ayuda: este problema tiene mucha semejanza con Independent Set. Recomendamos reducir dicho problema a Path Selection.