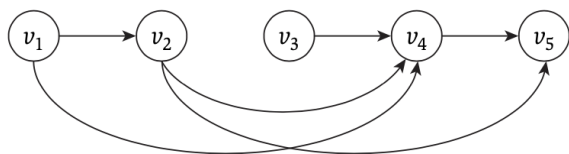


Examen parcial (2do recuperatorio) – 08/07/2024

- Definimos a un grafo ordenado como un grafo dirigido con vértices v_1, \dots, v_n en el que todos los vértices, salvo v_n tienen al menos una arista que sale del vértice, y cada arista va de un vértice de menor índice a uno de mayor índice (es decir, las aristas tienen la forma (v_i, v_j) con $i < j$). Implementar un algoritmo de **programación dinámica** que dado un grafo ordenado (y, si les resulta útil, una lista con los vértices en orden) determine cuál es la longitud del camino más largo. Dar la ecuación de recurrencia correspondiente. Dar también el algoritmo de reconstrucción de la solución. Indicar y justificar la complejidad del algoritmo implementado. Se pone a continuación un ejemplo de un grafo ordenado.



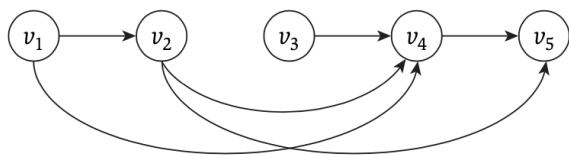
- La Cruz Roja cuenta con n ambulancias, de las cuáles conoce la ubicación de cada una. En un momento dado llegan p pedidos de ambulancias para socorrer personas. Debido a diferentes reglas que tienen, una ambulancia no debe trasladarse más de k kilómetros. Se quiere saber si se puede hacer una asignación de ambulancias a los pedidos, asignando cada una a como máximo 1 pedido. Implementar un algoritmo que resuelva este problema, utilizando **redes de flujo**. Indicar y justificar la complejidad del algoritmo implementado para el problema planteado.
- Implementar un algoritmo greedy que permita obtener el Independent Set máximo (es decir, que contenga la mayor cantidad de vértices) para el caso de un árbol (en el contexto de teoría de grafos, no un árbol binario). Indicar y justificar la complejidad del algoritmo implementado. Justificar por qué se trata de un algoritmo greedy. Indicar si el algoritmo siempre da solución óptima. Si lo es, explicar detalladamente, sino dar un contraejemplo.
- El problema de Subgrafo Isomórfico es el siguiente: *Dado un grafo G_1 y otro grafo G_2 , ¿existe un **subgrafo** de G_1 que sea isomorfo a G_2 ?* En el ejemplo del dorso, G_2 es isomorfo al subgrafo de G_1 que contiene a v_0, v_1, v_2 ; G_3 es isomorfo al subgrafo de G_1 que contiene a v_1, v_2, v_3 ; mientras que G_4 no es isomorfo a ningún subgrafo de G_1 .

Demostrar que el problema de Subgrafo Isomórfico es un problema NP-Completo. Recomendación: Recordar que el problema de K-Clique es un problema NP-Completo.

- Implementar un algoritmo que dado un grafo, obtenga el clique de mayor tamaño del mismo.

Examen parcial (2do recuperatorio) – 08/07/2024

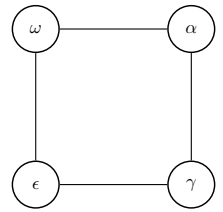
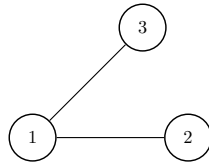
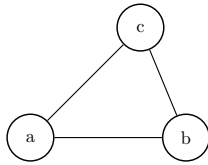
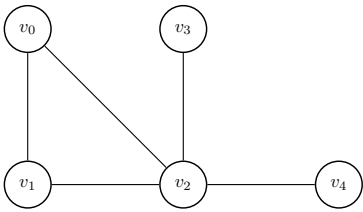
- Definimos a un grafo ordenado como un grafo dirigido con vértices v_1, \dots, v_n en el que todos los vértices, salvo v_n tienen al menos una arista que sale del vértice, y cada arista va de un vértice de menor índice a uno de mayor índice (es decir, las aristas tienen la forma (v_i, v_j) con $i < j$). Implementar un algoritmo de **programación dinámica** que dado un grafo ordenado (y, si les resulta útil, una lista con los vértices en orden) determine cuál es la longitud del camino más largo. Dar la ecuación de recurrencia correspondiente. Dar también el algoritmo de reconstrucción de la solución. Indicar y justificar la complejidad del algoritmo implementado. Se pone a continuación un ejemplo de un grafo ordenado.



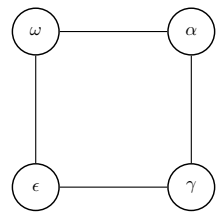
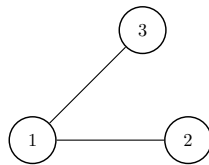
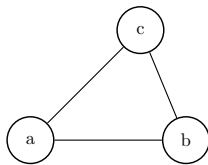
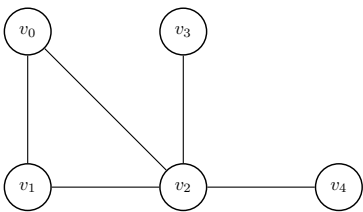
- La Cruz Roja cuenta con n ambulancias, de las cuáles conoce la ubicación de cada una. En un momento dado llegan p pedidos de ambulancias para socorrer personas. Debido a diferentes reglas que tienen, una ambulancia no debe trasladarse más de k kilómetros. Se quiere saber si se puede hacer una asignación de ambulancias a los pedidos, asignando cada una a como máximo 1 pedido. Implementar un algoritmo que resuelva este problema, utilizando **redes de flujo**. Indicar y justificar la complejidad del algoritmo implementado para el problema planteado.
- Implementar un algoritmo greedy que permita obtener el Independent Set máximo (es decir, que contenga la mayor cantidad de vértices) para el caso de un árbol (en el contexto de teoría de grafos, no un árbol binario). Indicar y justificar la complejidad del algoritmo implementado. Justificar por qué se trata de un algoritmo greedy. Indicar si el algoritmo siempre da solución óptima. Si lo es, explicar detalladamente, sino dar un contraejemplo.
- El problema de Subgrafo Isomórfico es el siguiente: *Dado un grafo G_1 y otro grafo G_2 , ¿existe un **subgrafo** de G_1 que sea isomorfo a G_2 ?* En el ejemplo del dorso, G_2 es isomorfo al subgrafo de G_1 que contiene a v_0, v_1, v_2 ; G_3 es isomorfo al subgrafo de G_1 que contiene a v_1, v_2, v_3 ; mientras que G_4 no es isomorfo a ningún subgrafo de G_1 .

Demostrar que el problema de Subgrafo Isomórfico es un problema NP-Completo. Recomendación: Recordar que el problema de K-Clique es un problema NP-Completo.

- Implementar un algoritmo que dado un grafo, obtenga el clique de mayor tamaño del mismo.



De izquierda a derecha, G_1, G_2, G_3, G_4



De izquierda a derecha, G_1, G_2, G_3, G_4