Trabajo Práctico n.º 2

PARTE 1

Clases de complejidad

Escribir el pseudocódigo de un algoritmo que resuelva cada uno de los siguientes problemas en tiempo polinomial, o bien demostrar que son NP-Completos.

1. Se tiene un conjunto de n actividades para seleccionar. Cada actividad tiene asociados un tiempo de inicio y tiempo de fin. Se dice que un conjunto de actividades es compatible si no hay dos que se superpongan en un tiempo. Se pide un algoritmo que devuelva verdadero o falso de acuerdo a si se puede encontrar un subconjunto compatible de tamaño k o superior.
2. Se tiene un conjunto de n actividades para seleccionar. Cada actividad tiene asociados un conjunto de tiempos de inicio y fin. Se dice que un conjunto de actividades es compatible si no hay dos que se superpongan en un tiempo. Se pide un algoritmo que devuelva verdadero o falso de acuerdo a si se puede encontrar un subconjunto compatible de tamaño k o superior.
3. En teoría de grafos, un camino hamiltoniano es un camino que visita cada vértice del grafo exactamente una vez. Se pide un algoritmo que indique si un grafo G tiene un camino hamiltoniano o no.
4. En teoría de grafos, un camino hamiltoniano es un camino que visita cada vértice del grafo exactamente una vez. Se pide un algoritmo que indique si un digrafo acíclico D tiene un camino hamiltoniano o no.
5. Se tiene un grafo dirigido y pesado G, cuyas aristas tienen pesos que pueden ser negativos. Se pide devolver verdadero o falso de acuerdo a si el grafo tiene algún ciclo con peso negativo.
6. Se tiene un grafo dirigido y pesado G, cuyas aristas tienen pesos que pueden ser negativos. Se pide devolver verdadero o falso de acuerdo a si el grafo tiene algún ciclo con exactamente igual a cero.

PARTE 2

Algoritmos de camino mínimo

Un grupo de conocidos inversores ha logrado modelar un dinámico sistema financiero usando un grafo dirigido. En este los vértices son modelados como monedas y sus aristas por unos valores relacionados a las tasas de cambio entre ellas, de manera tal que el camino mínimo entre dos monedas represente la manera óptima de cambiar dinero de una en otra.[1]

Dado que el conocimiento de estos financistas sobre algoritmia es limitado, acuden a estudiantes de Teoría de Algoritmos I para asesoramiento.

Se pide, entonces:

1. Escribir un breve informe (máximo cinco párrafos), explicando las ventajas y desventajas de los tres algoritmos de camino mínimo vistos en el curso (Dijkstra, Bellman-Ford y Floyd-Warshall). Explicar el principio de funcionamiento de cada uno y a qué técnica de programación responde.
2. El hecho de encontrar ciclos negativos en este problema significa que podríamos explotar el tipo de cambio para potencialmente ganar dinero infinito. Explicar cómo se identificaría esta situación usando los algoritmos descriptos.
3. Escribir un programa que dado un número n genere un digrafo completo de n vértices (y por lo tanto $n(n-1)$ aristas), con todos pesos positivos.
4. Implementar los algoritmos de Dijkstra, Bellman-Ford y Floyd-Warshall para encontrar caminos mínimos en grafos.
5. Extraer breves conclusiones (máximo dos párrafos) sobre el rendimiento de cada algoritmo ante diferentes tamaños de la entrada comparándolo con su complejidad computacional teórica.

1: Para analizar más en detalle este modelo, consultar “Algorithms”, R. Sedgewick, K. Wayne, (cuarta edición), cap. 4.4 “Arbitrage”, pp: 679-681.

Aclaraciones generales

1. El informe de todo el trabajo no debe superar las cuatro carillas de extensión, y deberá incluir las instrucciones para ejecutar todos los programas desarrollados.
2. Para la implementación de los algoritmos se podrán usar todo tipo de bibliotecas, excepto de grafos.
3. Los grafos deberán ser generados en el formato propuesto por Sedgewick, en donde los vértices estarán nombrados según indentificadores desde 0 hasta $V-1$, y los archivos de están confeccionados según:
   - Una primera línea indicando la cantidad de vértices V.
   - Una segunda línea indicando la cantidad de aristas A.
   - Sucesivas líneas representando cada arista, en dónde se indican los vértices de origen y destino, separados por espacios. Sólo se listará una de las direcciones si el grafo es no dirigido.