Recuperatorio de Programación 3 - TUDAI 9/11/2024

Ejercicio 1

a) Dado un grafo dirigido G (V, A) **implementar en Java** un algoritmo que determine **si es posible** ir desde un vértice A a otro vértice B y luego regresar de B hasta A.

b) Si se debieran encontrar, de forma eficiente, los caminos más cortos desde todos los vértices de un Grafo dirigido a un vértice dado X ¿Qué algoritmo utilizaría? Justificar la respuesta.
Nota: Tener en cuenta si se debe cumplir alguna condición para que se pueda aplicar el algoritmo elegido.

Ejercicio 2

Muestre en JAVA la **declaración de la clase Lista** doblemente vinculada, sus **atributos de instancia** y la **implementación eficiente** de sólo los siguientes métodos. Luego indique la **complejidad resultante** de cada uno de ellos.

public void agregarPrincipio(T elemento)
public void agregarFinal(T elemento)
public bool eliminarElemento(T elemento) //Indica si pudo o no eliminar el elemento indicado.
public void eliminarUltimo()

Suponga que la clase Nodo<T> ya está implementada con los métodos usuales (next, prev y value).

Ejercicio 3

Dado el siguiente problema que se quiere resolver mediante la técnica Greedy, responda:

a) Cuál sería la **estrategia Greedy** que seguiría,
b) Plantear un **algoritmo java** que lo resuelva.

En un call center se reciben un conjunto P de distintas consultas que vienen acompañadas de un tiempo de atención estimado. El call center dispone de un grupo de operadores donde cada uno de ellos podrá dedicar un tiempo X por día a atender dichas consultas. Algunas de estas consultas son administrativas, mientras que otras son consultas técnicas. El problema es que los operadores detestan contestar consultas técnicas, por lo que la empresa impuso un límite: a cada operador se le podrá asignar, como máximo, M consultas técnicas por día.
Se desea construir un algoritmo que, dado el conjunto P de consultas para un día, intente determinar el número mínimo de operadores que serán requeridos para atenderlas.
Nota: Suponga que existe el método auxiliar "*boolean esConsultaTecnica(consulta)*"

Ejercicio 4

Se tiene un **conjunto de N elementos** y se quieren obtener todas las formas distintas de colocar esos elementos en una secuencia S.
Por ejemplo, para el conjunto {1, 2, 3} se deben obtener las siguientes secuencias {1, 2, 3}, {1, 3, 2}, {2, 1, 3}, {2, 3, 1}, {3, 1, 2}, {3, 2, 1}.

a) Dibuje el árbol de exploración del algoritmo, indicando qué decisiones se toman en cada paso y qué información se lleva en los estados.
b) **Escriba un algoritmo java** que resuelva el problema mediante **Backtracking.**

Ejercicio 5

Construir la **tabla de seguimiento de Dijkstra** para el grafo que se presenta a continuación sabiendo que el vértice B se toma como origen. También presentar el **árbol de caminos mínimos** resultante.