Definición de los TDAs a utilizar en todo el TPO

ColaCadena

Cola *es una estructura que permite almacenar una colección de cadenas de caracteres (Strings), de modo tal que el elemento más antiguo sea el primer elemento en salir. También se la conoce como una estructura FIFO.*

Operaciones básicas:

InicializarCola: inicializa la estructura cola.
Acolar: agrega una cadena a la cola. La estructura debe estar inicializada.
Desacolar: elimina la cadena más antigua agregada. La estructura no debe estar vacía.
Primero: obtiene la primera cadena a eliminar (no se elimina). La estructura no debe estar vacía.
ColaVacia: indica si la cola no contiene cadenas. La estructura debe estar inicializada.

Tabla

Tabla *es una estructura que permite almacenar una colección de elementos. Cada elemento consta de un nombre asociado a un código (tipo de dato entero).*

Operaciones básicas:

Inicializartabla: inicializa la tabla.
Agregar: esta operación agrega un elemento nuevo a la tabla, donde el código corresponde a la cantidad de elementos de la tabla antes de ingresar un nuevo nombre. El nombre no debe existir previamente en la tabla. La tabla debe estar inicializada. Al no existir las operaciones *eliminarElemento* ni *actualizarElemento*, los códigos serán únicos y consecutivos.
Pertenece: esta operación permite saber si un nombre ya fue ingresado a la tabla. La tabla debe estar inicializada.
Codigo: indica cuál es el código de un nombre suministrado (no se elimina). El nombre debe existir.
Tabla: devuelve los elementos en una estructura cola, donde cada elemento de la cola corresponde a la concatenación del código y el nombre separado por el carácter ";", según el orden en que se encuentren guardados. La tabla debe estar inicializada.
OrdenarNombres: ordena los elementos de la tabla, alfabéticamente por los nombres. La tabla debe estar inicializada.
OrdenarCodigos: ordena los elementos de la tabla por los códigos, en forma ascendente. La tabla debe estar inicializada.
EstaVacia: informa si la tabla no tiene elementos. La tabla debe estar inicializada.

Pila

Pila *es una estructura que permite almacenar una colección de valores enteros de modo tal que el último elemento en ingresar sea el primer elemento por salir. También se la conoce como una estructura LIFO.*

Operaciones básicas:

InicializarPila: inicializa la estructura pila.
Apilar: agrega un elemento. La estructura debe estar inicializada.
Desapilar: elimina el último elemento agregado. La estructura no debe estar vacía.
Tope: obtiene el primer elemento a eliminar (no se elimina). La estructura no debe estar vacía.
PilaVacia: indica si la pila no contiene elementos. La estructura debe estar inicializada.

Cola

Cola *es una estructura que permite almacenar una colección de valores enteros, de modo tal que el elemento más antiguo sea el primer elemento en salir. También se la conoce como una estructura FIFO.*
Operaciones básicas:
InicializarCola: inicializa la estructura cola.
Acolar: agrega un elemento a la cola. La estructura debe estar inicializada.
Desacolar: elimina el elemento más antiguo agregado. La estructura no debe estar vacía.
Primero: obtiene el primer elemento a eliminar (no se elimina). La estructura no debe estar vacía.
ColaVacia: indica si la cola no contiene elementos. La estructura debe estar inicializada.

Cola con prioridad

Cola con prioridad *es una estructura que permite almacenar una colección de valores enteros asociados a una prioridad, de modo tal que el primer elemento a salir sea el de mayor prioridad.*
Operaciones básicas:
InicializarCola: inicializa la estructura cola.
AcolarPrioridad: agrega un elemento 'x' con prioridad 'p', ambos datos suministrados. La estructura debe estar inicializada.
Desacolar: elimina el elemento con mayor prioridad. La estructura no debe estar vacía.
Primero: obtiene el valor del dato de mayor prioridad. El valor no se elimina. La estructura no debe estar vacía.
Prioridad: obtiene la prioridad del dato de mayor prioridad (no se elimina). La estructura no debe estar vacía.
ColaVacia: indica si la cola no contiene elementos. La estructura debe estar inicializada.

Conjunto

Conjunto *es una estructura que permite almacenar una colección de valores enteros, no repetidos y no necesariamente ordenado.*
Operaciones básicas:
InicializarConjunto: inicializa el conjunto.
Agregar: agrega un elemento suministrado. El conjunto debe estar inicializado y el elemento no debe existir.
Sacar: elimina un elemento suministrado. EL elemento debe pertenecer al conjunto.
Obtener: devuelve un valor cualquiera del conjunto. El elemento no se elimina. El conjunto no debe estar vacío.
ConjuntoVacio: devuelve verdadero si el conjunto no tiene elementos. El conjunto debe estar inicializado.
Pertenece: devuelve verdadero si el valor suministrado pertenece al conjunto. El conjunto debe estar inicializado.

Diccionario simple

Diccionario simple *es una colección de pares asociados clave-valor, ambos números enteros. Las claves son únicas. No puede existir clave sin valor asociado.*
Operaciones básicas:
InicializarDiccionarioSimple: inicializa el diccionario.
Agregar: permite agregar una clave con un valor, ambos suministrados. El diccionario debe estar inicializado y la clave no debe existir.

Eliminar: permite eliminar una clave suministrada. La clave debe existir.
Obtener: devuelve el valor asociado a una clave suministrada. No elimina el valor. La clave debe existir.
Claves: devuelve el conjunto de claves del diccionario. No elimina las claves. El diccionario debe estar inicializado

Diccionario múltiple

Diccionario múltiple *es una colección de elementos asociados clave-valores. La clave y los valores son números enteros. Las claves son únicas. No puede existir clave sin valor asociado. Los valores son únicos para una misma clave.*
Operaciones básicas:
InicializarDiccionarioMultiple: inicializa el diccionario.
Agregar: agrega un valor a una clave, ambos datos suministrados. El diccionario debe estar inicializado y el valor no debe existir para esa clave. Si no existe la clave, se agrega.
Eliminar: elimina una clave suministrada junto con los valores asociados. La clave debe existir.
EliminarValor: elimina un valor asociado a una clave, ambos datos suministrados. Ambos deben existir.
Obtener: devuelve el conjunto de valores asociados a una clave suministrada. No elimina los valores. La clave debe existir.
Claves: devuelve el conjunto de claves del diccionario. No elimina las claves. El diccionario debe estar inicializado.

Árbol binario de búsqueda (ABB)

Árbol de búsqueda binaria *es una colección de elementos ordenados en forma jerárquica (todos números enteros). Entre dos elementos vinculados se define la relación padre-hijo. Cada elemento tiene sólo un padre, excepto el elemento raíz del árbol que no tiene padre.*
Operaciones básicas:
InicializarABB: permite inicializar el árbol.
Raiz: devuelve el valor de la raíz. No elimina el elemento. El árbol no puede estar vacío.
HijoIzq: devuelve la referencia al subárbol izquierdo. No elimina los elementos del sub-árbol. El árbol no puede estar vacío.
HijoDer: devuelve la referencia al subárbol derecho. No elimina los elementos del sub-árbol. El árbol no puede estar vacío.
Agregar: agrega un valor suministrado. El árbol debe estar inicializado y no debe existir el valor.
Eliminar: elimina un valor suministrado. El valor debe existir.
ArbolVacio: devuelve verdadero si el árbol no contiene elementos. La estructura debe estar inicializada.
Pertence: devuelve verdadero si el valor suministrado pertenece al árbol. EL árbol debe estar inicializado.

Grafo

Grafo *es una colección de vértices etiquetados con un valor entero y aristas con un peso asociado, definidas por un vértice origen y un vértice destino. El grafo no puede tener aristas paralelas ni bucles. Los pesos de las aristas son números enteros.*
Operaciones básicas:
InicializarGrafo: inicializa el grafo.
AgregarVertice: agrega un vértice. El grafo debe estar inicializado y el vértice no debe existir.
EliminarVertice: elimina un vértice suministrado. El vértice debe existir.
AgregarArista: agrega una arista identificada por los vértices de origen y destino y el peso. Los vértices deben existir, pero no debe existir la arista.

EliminarArista: elimina una arista. Se suministran los vértices de origen y destino. La arista debe existir.
Vertices devuelve el conjunto de vértices de. No elimina los vértices. El grafo debe estar inicializado
Peso: devuelve el peso de la arista, identificada por los vértices de origen y destino. La arista debe existir
ExisteArista: devuelve verdadero si la arista, identificada por los vértices de origen y destino, existe. El grafo debe estar inicializado.