Algoritmos y Estructuras de Datos II - 1° cuatrimestre 2020

Práctico 2 - Parte 2

1. Completá la implementación de listas dada en el teórico usando punteros.

2. Dada una constante natural N, implementá el TAD Lista de elementos de tipo T, usando un arreglo de tamaño N y un natural que indica cuántos elementos del arreglo son ocupados por elementos de la lista. ¿Esta implementación impone nuevas restricciones? ¿En qué función o procedimiento tenemos una nueva precondición?

3. Implementá el procedimiento *add_at* que toma una lista de tipo T, un natural n, un elemento e de tipo T, y agrega el elemento e en la posición n, quedando todos los elementos siguientes a continuación. Esta operación tiene como precondición que n sea menor al largo de la lista.

   AYUDA: Puede ayudarte usar las operaciones copy, take y drop.

4. (a) Especificá un TAD *tablero* para mantener el tanteador en contiendas deportivas entre dos equipos (equipo A y equipo B). Deberá tener un constructor para el comienzo del partido (tanteador inicial), un constructor para registrar un nuevo tanto del equipo A y uno para registrar un nuevo tanto del equipo B. El tablero sólo registra el estado actual del tanteador, por lo tanto el orden en que se fueron anotando los tantos es irrelevante.

   Además se requiere operaciones para comprobar si el tanteador está en cero, si el equipo A ha anotado algún tanto, si el equipo B ha anotado algún tanto, una que devuelva verdadero si y sólo si el equipo A va ganando, otra que devuelva verdadero si y sólo si el equipo B va ganando, y una que devuelva verdadero si y sólo si se registra un empate.

   Finalmente habrá una operación que permita anotarle un número n de tantos a un equipo y otra que permita "castigarlo" restándole un número n de tantos. En este último caso, si se le restan más tantos de los acumulados equivaldrá a no haber anotado ninguno desde el comienzo del partido.

   (b) Implementá el TAD Tablero utilizando una tupla con dos contadores: uno que indique los tantos del equipo A, y otro que indique los tantos del equipo B.

   (c) Implementá el TAD Tablero utilizando una tupla con dos naturales: uno que indique los tantos del equipo A, y otro que indique los tantos del equipo B. ¿Hay alguna diferencia con la implementación del inciso anterior? ¿Alguna operación puede resolverse más eficientemente?

5. Especificá el TAD Conjunto finito de elementos de tipo T. Como constructores considerá el conjunto vacío y el que agrega un elemento a un conjunto. Como operaciones: una que chequee si un elemento e *pertenece* a un conjunto c, una que chequee si un conjunto *es vacío*, la operación de *unir* un conjunto a otro , *insersectar* un conjunto con otro y obtener la *diferencia*. Estas últimas tres operaciones deberían especificarse como procedimientos que toman dos conjuntos y modifican el primero de ellos.

6. Implementá el TAD Conjunto finito de elementos de tipo T utilizando:

   (a) una lista de elementos de tipo T, donde el constructor para agregar elementos al conjunto se implementa directamente con el constructor **addl** de las listas.

   (b) una lista de elementos de tipo T, donde se asegure siempre que la lista está ordenada crecientemente y no tiene elementos repetidos. Debes tener cuidado especialmente con el constructor de agregar elemento y las operaciones de unión, intersección y diferencia. A la propiedad de mantener siempre la lista ordenada y sin repeticiones le llamamos *invariante de representación*.

   Ayuda: Para implementar el constructor de agregar elemento puede serte muy útil la operación *add_at* implementada en el punto 3.