

ANÁLISIS Y DISEÑO DE ALGORITMOS I

Práctico N° 3. Tipos de Datos Abstractos

1. Especifique algebraicamente el TDA *Natural* con la siguiente funcionalidad básica: *cero*, *sucesor*, *predecesor*, *+*, *=*.
 2. Especifique algebraicamente los TDAs *Persona*, *Libro*. Implemente en C++ los archivos header.
 3. Especifique algebraicamente los TDAs *Punto*, *Segmento* y *Círculo*. Implemente en C++ los archivos header.
 4. Para cada uno de los siguientes Tipos de Datos Abstractos:
 - Escriba una especificación algebraica.
 - Clasifique sus operaciones según su signatura en constructoras, modificadoras u observadoras.
 - Escriba algunos términos del álgebra.
 - Reescriba los términos del inciso anterior en función de las constructoras básicas.
 - Discuta las distintas alternativas de implementación propuestas en cada inciso en base a la complejidad temporal de sus funciones. Seleccione una de ellas e implemente.
 - a) TDAs *conjunto*, *lista*, *pila* y *cola*.
Posibles representaciones:
 - arreglo
 - listas vinculadas
 - b) TDA *árbol binario de Búsqueda*.
Posibles representaciones:
 - arreglo
 - punteros (hijo izquierdo, hijo derecho).
 - c) TDA *árbol n-ario*.
Posibles representaciones:
 - arreglo con punteros al padre
 - arreglo de punteros a hijos,
 - hijo más a la izquierda, hermano derecho.
 - punteros a listas de hijos (listas vinculadas)
 - d) TDA *heap*.
Implemente utilizando la representación basada en arreglo.
 - e) TDA *Polinomio* de una variable con coeficientes enteros,
$$p(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1} + a_n x^n, \quad n \in \mathbb{N}, \quad \forall i : 0 \leq i \leq n: a_i \in \mathbb{Z},$$

con las siguientes operaciones:
 - *cero* : representa el polinomio $p(x) = 0$
 - *añadir*: añade un término (coeficiente-exponente) a un polinomio
 - *evaluar*: calcula el valor del polinomio en un punto
 - *coeficiente*: devuelve el coeficiente asociado a un término de exponente dado dentro de un polinomio
 - *suma*: dado dos polinomios, devuelve la suma de los mismos.
 5. Analice la funcionalidad del tipo *String* provisto por C++. Abstraiga la funcionalidad básica y especifique en NEREUS.
-