Profesores: Leandro LLanza
Roberto Konow

ESTRUCTURA DE DATOS – SOLEMNE 1

Pregunta 1

Un documento en HTML puede ser visto como un conjunto de "tags", que permiten agregar diferentes tipos de contenido con su respectivo formato. Un tag o etiqueta es una instrucción que está delimitadas por corchetes angulares (<,>).

En un análisis para este tipo de documentos se necesita conocer:

1. La cantidad total de tags válidos que contiene un documento,
2. La cantidad de comentarios que posee el documento y
3. La cantidad de imágenes que posee el documento.

Un comentario: `<!-- este texto es una comentario ... -->`
Una imagen: `<img ...>`

Tags válidos: `<a> <a b c> </a> <xyz ... zyx> <i> <img src="wse.gif"> <!- - inicio - -> < >`
Tags inválidos: `<a<as> <<img>> <xd<>`

a) Diseñe **una** máquina de estados (dibuje el diagrama) que resuelva los tres problemas anteriores. (15 puntos)

b) Implemente un programa que represente **fidedignamente** su diagrama de estados. (15 puntos)

Pregunta 2 (bonus)

Se desea manejar una lista ordenada en forma ascendente de un conjunto de elementos, todos distintos, para ello utilizaremos la siguiente estructura.

```
class ALN {       // Ascending List Node
      int data;
      ALN next;
}
```

Se le pide implementar el método

```
      void updateSumaLista(ALN head, int x, int delta)
```

,que busca el dato `x` en la lista `head`, le suma `delta` y mantiene el orden ascendente. Si se da el caso de elementos repetidos, elimina uno de ellos. `delta` es siempre positivo. Si desea, puede implementar esta lista utilizando un nodo cabecera sin datos. (10 puntos)

Pregunta 3

Un polinomio puede ser representado como:

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_2 x^2 + a_1 x + a_0$$

Donde a_i corresponde a una constante de tipo entero y x es la variable

Se le pide a usted que defina e implemente un TDA (Tipo de Dato Abstracto) que sea capaz de representar y manipular un polinomio cualquiera. (30 puntos)
El TDA debe soportar las siguientes operaciones:

`void imprime()`
Imprime el polinomio en la forma antes mostrada (de mayor a menor potencia).

`int evalua(int x)`
Retorna el resultado de evaluar el polinomio.

`Polinomio deriva()`
Retornar un nuevo polinomio con la derivada del original.

`void deriva()`
Deriva el polinomio.

`int grado()`
Retorna el grado del polinomio.

`boolean esIgual(Polinomio p)`
Retorna Verdadero si el polinomio original es igual al polinomio p. Dados dos polinomios de igual grado, se dice que son iguales si los coeficientes de los monomios de igual grado son iguales. Por ejemplo $Q(x) = 5x + 5x^3 - 4 - x^2$ y $P(x) = 5x^3 - x^2 + 5x - 4$, son iguales.

`void multiplicaEscalar(int c)`
Multiplica las constantes a_i por el escalar c.

`void divideEscalar(int c)`
Divide las constantes a_i por el escalar c.

`Polinomio suma(Polinomio p)`
Retorna un nuevo polinomio con la suma del polinomio original más el polinomio p. Ejemplo:

$$\begin{array}{rccccccc} & 3x^6 & -2x^5 & +8x^4 & +8x^3 & -3x^2 & +7x & +1 \\ + & & +4x^5 & +x^4 & +9x^3 & -12x^2 & +6x & -5 \\ \hline & 3x^6 & +2x^5 & +9x^4 & +17x^3 & -15x^2 & +13x & -4 \end{array}$$

`void suma(Polinomio p)`
Suma al polinomio original el polinomio p.