SESIÓN **/12**

RECURSIVIDAD

1. Recursividad en Estructuras Estáticas
2. Recursividad en Estructuras Dinámicas

**INTRODUCCIÓN**

* La clase anterior conocimos los procedimientos recursivos. Describimos la forma cómo aplicarlos y analizamos su eficiencia en comparación con los procedimientos iterativos.
* En la presente sesión conoceremos la forma de implementar algoritmos recursivos en Arreglos unidimensionales y en Listas enlazadas simples.
* Culminaremos comparando la eficiencia de los algoritmos recursivos versus los algoritmos iterativos, aplicados en Estructuras de Datos.

1. **RECURSIVIDAD EN ESTRUCTURAS ESTÁTICAS**

Cuando se aplica recursividad en las operaciones sobre un Arreglo, el método recursivo recibe la posición de un elemento. Cada vez que se invoca al método recursivo, el valor que se le envía puede aumentar en 1 ó disminuir en 1 respecto a la llamada anterior.



***EJEMPLO 1:***

***Procedimiento recursivo para mostrar los elementos de un Vector (del último al primero).***

*public void mostrar (int i)*

*{*

*if (i >= 0)*

*{*

*// mostrar elemento[i]*

*mostrar (i – 1);*

*}*

*}*

***Cada vez que se invoca al método recursivo, el valor de i disminuye en 1.***

***La primera vez que se invoca al método recursivo, se le envía la posición del último elemento.***

***EJEMPLO 2:***

***Procedimiento recursivo para mostrar los elementos de un Vector (del primero al último).***

*public void mostrar (int i)*

*{*

*if (i < conta)*

*{*

*// mostrar elemento[i]*

*mostrar (i + 1);*

*}*

*}*

***Cada vez que se invoca al método recursivo, el valor de i aumenta en 1.***

***La primera vez que se invoca al método recursivo, se le envía la posición del primer elemento.***

***EJEMPLO 3:***

***Procedimiento recursivo para buscar un número dentro de un Vector (del último al primero).***

*public int buscar (int i, double dato)*

*{*

*if (i < 0)*

*return -1;*

*else*

*if (elemento[i] == dato)*

*return i;*

*else*

*return buscar (i – 1, dato);*

*}*

***Cada vez que se invoca al método recursivo, el valor de i disminuye en 1.***

***La primera vez que se invoca al método recursivo, se le envía la posición del último elemento y el número a buscar.***

1. **RECURSIVIDAD EN ESTRUCTURAS DINÁMICAS**

Cuando se aplica recursividad en las operaciones sobre una Lista enlazada, el método recursivo recibe la dirección de memoria de un elemento. Cada vez que se invoca al método recursivo, se le envía la dirección de memoria del elemento anterior ó la dirección de memoria del siguiente elemento.

inicio

null

dato

dato

dato

dato

***EJEMPLO 1:***

***Procedimiento recursivo para mostrar los elementos de una Lista enlazada simple.***

*public void mostrar (Nodo p)*

*{*

*if (p != null)*

*{*

*// mostrar p.info*

*mostrar (p.sgte);*

*}*

*}*

***Cada vez que se invoca al método recursivo, la variable p apunta al siguiente elemento de la Lista.***

***La primera vez que se invoca al método recursivo, se le envía la dirección de memoria dónde está ubicado el primer elemento.***

***EJEMPLO 2:***

***Procedimiento recursivo para buscar un número dentro de una Lista enlazada simple.***

*public Nodo buscar (Nodo p, double dato)*

*{*

*if (p == null)*

*return null;*

*else*

*if (p.info == dato)*

*return p;*

*else*

*return buscar (p.sgte, dato);*

*}*

***Cada vez que se invoca al método recursivo, la variable p apunta al siguiente elemento de la Lista.***

***La primera vez que se invoca al método recursivo, se le envía la dirección de memoria dónde está ubicado el primer elemento y el número a buscar.***

**CONCLUSIÓN**

En general, los procedimientos recursivos recorren un Vector del último al primer elemento. Por el contrario, los procedimientos recursivos recorren una Lista enlazada simple del primer al último elemento. Tratándose de Arreglos y Listas enlazadas, resultan más eficiente los algoritmos iterativos en comparación con los algoritmos recursivos.

**REFERENCIAS BIBLIOGRÁFICAS**

* Cairo, O.; Guardati, S. (2008). Estructuras de datos. 3ra. Edición. México D.F., Mexico: McGraw Hill.
* Instituto NIIT (2011). Data Structures and Algorithms. Student guide.