Auxiliar 2 - CC3301

Alonso González Ulloa

April 23, 2010

Problema 1

Vea P3 de la clase anterior.

Problema 2 [C1 - 2005]

Escriba una función recursiva que construya un árbol binario de búsqueda (los hijos izquierdos siempre tienen elementos menores al nodo y los hijos derechos siempre tienen elementos mayores).

La función debe retornar un árbol binario nuevo (pedido con malloc) que contiene los elementos de un arreglo ordenados de mayor a menor. La función es:

```
typedef struct {
    int val;
    tree *left, *right,
} tree;

tree *make_tree(int *a, int i, int j);
```

La función usa los elementos entre $a[i]$ y $a[j]$ (inclusivos). Por lo que la primera llamada para obtener todos los elementos de un arreglo con n elementos es:

```
int a[N];

t = make_tree(a, 0, N-1);
```

La lógica de la función es siempre dejar como raíz del árbol al elemento que está en la mitad del arreglo y dejar en el sub-árbol izquierdo los elementos anteriores a ese y en el sub-árbol derecho a los posteriores.

Pregunta 3[C1 - 2005]

Para esta pregunta no puede usar funciones de biblioteca de C, salvo malloc.

Escriba una función que busca un patrón dentro de un string. El patrón puede calzar en forma parcial y en ese caso debemos retornar el calce más largo que figura en el string. El string retornado, que es substring del patrón, debe estar en memoria nueva (pedida con malloc). Si no se encuentra ningún calce de ningún tipo, debe retornar el string vacío.

```
char *substring(char *patrón, char *string);
```

Ejemplos:

```
substring("arm", "arista");  
--> "ar"  
substring("arm", "arto armado");  
--> "arm"
```