

Ejercicios de primer parcialito

Taller 14/09

Del primer parcialito del segundo cuatrimestre de 2012

1) Escribir una función que reciba una pila y un parámetro n con la cantidad de elementos en la pila y la “parta” en dos: La que es recibida por parámetro contendrá la primera mitad (elementos de 0 a $n/2-1$), y la pila devuelta por la función tendrá los elementos de $n/2$ a $n-1$, manteniendo el orden de la pila original. (Se pueden usar las estructuras auxiliares que crean convenientes)

Del primer recuperatorio del parcialito del segundo cuatrimestre de 2012

2) Escriba una función recursiva para calcular de forma eficiente la potencia de un número con exponente entero positivo.

3) a) Implementar la función `void invertirCola(cola_t* cola)`, que invierte el orden de sus elementos.

b) Indicar el orden de complejidad de la función implementada.

Del primer recuperatorio del primer parcialito, segundo cuatrimestre de 2013

4) Escribir una primitiva que invierta el contenido de una *Lista Enlazada* sin usar ninguna estructura auxiliar.

5) El StoogeSort es un algoritmo de ordenamiento basado en Los Tres Chiflados que funciona de la siguiente forma:

- Si el primer elemento es más grande que el ultimo, los intercambia.
- Si la cantidad de elementos del vector es mayor o igual a 3:
 1. Llama a StoogeSort recursivamente con los primero dos tercios del vector (desde 0 hasta $2/3 * cantidad$)
 2. Llama a StoogeSort recursivamente con los segundos dos tercios (desde $1/3 * cantidad$, hasta cantidad)
 3. Vuelve a llamar recursivamente a StoogeSort con los primeros dos tercios del vector.

Utilizando el teorema maestro, calcular el orden del algoritmo, suponiendo que la operación de partir es de tiempo constante.

Del primer recuperatorio del primer parcialito, primer cuatrimestre de 2014

6) Se pide implementar una función recursiva que calcule el largo de una pila sin utilizar estructuras auxiliares ni acceder a su estructura interna (que no sabemos cuál es). El prototipo de la función es:

```
size_t largo_pila(pila_t *pila);
```

Se puede llamar a cualquier primitiva de la pila, incluyendo aquellas que la modifiquen, pero siempre la pila debe quedar en su estado original al terminar la ejecución de la función.

Del primer parcialito, primer cuatrimestre de 2016

7) Se tiene un arreglo tal que $[1, 1, 1, \dots, 0, 0, \dots]$ (es decir, “unos seguidos de ceros”). Se pide:
a) una función de orden $O(\log n)$ que encuentre el índice del primer 0. Si no hay ningún 0 (solo hay unos), debe devolver -1.
b) demostrar con el Teorema Maestro que la función es, en efecto, $O(\log n)$.

Ejemplos:

```
[1, 1, 0, 0, 0] ->2  
[0, 0, 0, 0, 0] ->0  
[1, 1, 1, 1, 1] ->-1
```

Del primer recuperatorio segundo parcialito, primer cuatrimestre de 2016

8) Implementar la primitiva de la lista `lista_t* lista_map(const lista_t*, void* (*f)(void*))` que dada una lista enlazada devuelva una nueva lista. Esta nueva lista debe contener al resultado de aplicar la función `f` pasada por parámetro a cada uno de los elementos de la lista original sin usar iteradores.
Indicar el orden de la solución propuesta.