

# Programación Avanzada – Algorítmica

## Tarea

Fecha de entrega: 13 de mayo, 2015

1. Jorge tiene un árbol binario de 26 nodos. Cada nodo está etiquetado con una letra única del alfabeto. Las secuencias de recorrido en pre-orden y post-orden del árbol son las siguientes:

pre-orden: M N H C R S K W T G D X I Y A J P O E Z V B U L Q F

post-orden: C W T K S G R H D N A O E P J Y Z I B Q L F U V X M

Dibuja el árbol binario de Jorge.

2. Da un algoritmo que encuentre el *segundo* entero más pequeño entre  $n$  enteros en a lo más  $n + \lceil \lg n \rceil - 2$  comparaciones. Tip: aplica una estrategia divide-and-conquer para encontrar al más chico, ¿dónde está el segundo más pequeño?
3. Dada una gráfica **no-dirigida**  $G = (V, E)$ , da un algoritmo que encuentre un ciclo en la gráfica, que visite cada arista *exactamente* una vez, o que diga que no es posible hacerlo.
4. Supón una gráfica  $G$  y un árbol mínimo generados (MST) de esa gráfica (esto es, el MST ya ha sido construido).
  - (a) Da un algoritmo para actualizar el MST cuando agregamos una arista a  $G$ .
  - (b) Da un algoritmo para actualizar el MST cuando una arista es borrada de  $G$ .
  - (c) Da un algoritmo para actualizar el MST cuando se agrega un vértice (y posiblemente aristas) a  $G$ .
5. Supón tener un apuntador a la cabeza de una lista simplemente ligada. Normalmente, cada nodo en la lista tiene solo un apuntador al elemento siguiente y el apuntador en el último nodo es `NULL`. Desafortunadamente, tu lista puede haber sido corrupta por un bug en el código de alguien más, de tal forma que el último nodo tiene un apuntador que regresa a otro nodo en lugar de `NULL` (Fig. 1). Describe un algoritmo que determine si la lista ligada esta corrupta o no. Tu algoritmo no debe modificar la lista. Intenta que se pueda ejecutar en tiempo  $O(n)$ , donde  $n$  es el número de nodos en la lista, y use  $O(1)$  espacio extra (sin contar la lista misma).

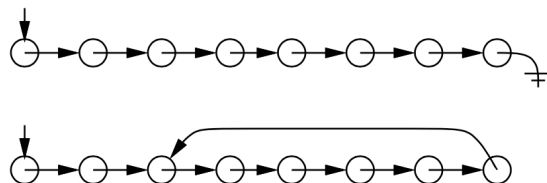


Figure 1: Arriba: Una lista ligada estándar. Abajo: Una lista ligada corrupta.

6. Repasar:

- Notación asintótica
- Estructuras de datos básicas.
- Algoritmos de ordenamiento.
- Conjuntos Disjuntos.
- Árboles: recorridos
- Gráficas: recorridos, MST, caminos más cortos.