

Auxiliar 6 - Listas enlazadas

Profesores: Nelson Baloian, Patricio Poblete
Benjamín Bustos
Sebastian Ferrada

Auxiliares: Vicente Olivares, Ricardo Valdivia
Sebastián Acuña
Valentina Aravena

P1. Reversar Lista de enlace simple (Basado en Ejercicio 4.3)

Considere la definición de la clase Lista descrita en los apuntes. Se pide que escriba un método que pueda ser invocado como `L.reversar()`, que al ejecutarse re-enlace los nodos de la lista de modo que queden en el orden opuesto al original, en tiempo lineal en el largo de la lista. Esto debe hacerse solo modificando referencias (o punteros), **sin crear nodos nuevos**.

Pruebe su definición con los siguientes casos:

- Lista de varios elementos
- Lista vacía
- Lista con un único elemento

P2. Mezclar listas enlazadas

Diseñe un algoritmo iterativo y escriba el método

```
mezclarListasOrdenadas(self, lista2)
```

Este se llamará desde una lista ordenada y recibirá como parámetro una segunda lista ordenada (ascendentemente por *valor*), retornará una lista que representará la mezcla ordenada de ambas listas.

P3. Intersectar listas enlazadas

Suponga que se tiene dos listas enlazadas, con cabecera, cuyos punteros a sus respectivas cabeceras son **a** y **b**, y que contienen números enteros ordenados en orden ascendente

Usted debe escribir una función **intersecta(a,b)** que reciba los punteros a las respectivas cabeceras y que construya y retorne una nueva lista con cabecera, también ordenada ascendentemente, en la cual aparezcan solamente los elementos que estaban en ambas listas **a** y **b**. Su función debe correr en tiempo Θ (suma de los largos de las dos listas).

Si le simplifica la vida, puede suponer que cada lista termina con un dato 999999, que se supone que es mayor que cualquier dato válido. En ese caso, la lista que usted construya también debe tener ese dato extra al final. Indique claramente si usted hace uso de esta suposición, colocando un comentario al inicio de su función.