

Actividad 0

Luis Miguel Fros De Castro

20 de Agosto de 2018

1. Complejidades Teóricas

Sea $L \in \mathcal{L}_{\mathbb{N}}$ ($\mathcal{L}_{\mathbb{N}}$ conjunto de listas ligadas) una lista ligada con un largo finito n . ($||L|| = n, n \in \mathbb{N}$). Para el análisis teórico de la complejidad de las operaciones, se asume que además de la referencia el primer elemento tenemos un puntero al ultimo.

- **Insert**(L, x, i): Insertar un elemento x en la posición i de L toma en el peor caso n pasos si $i=n$, por lo debe recorrer todos los elementos de la lista.
 - Complejidad es $O(n)$.
- **Delete**(L, i): Eliminar un elemento en la posición i de L también debe recorrer todos los elementos de la lista hasta estar en la posición buscada. En el peor de los casos $i=n$.
 - Complejidad es $O(n)$.
- **Append**(L, x): Dada la implementación mencionada, esta operación sería actualizar el ultimo nodo como x .
 - Complejidad es $O(1)$.
- **Pop**(L): En este caso, aunque tengamos la referencia al ultimo elemento necesitamos su predecesor. Esto requiere recorrer $n-1$ elementos hasta encontrarlo.
 - Complejidad es $O(n)$.

Sea $L* \in \mathcal{L}_{\mathbb{N}}$ de largo k .

- **Concatenate**($L, L*$): Dado que tenemos la referencia al ultimo nodo, esta operación es equivalente a Append dado que el ultimo elemento de L tiene como proximo al primero de $L*$.
 - Complejidad es $O(1)$
- **Destroy**(L): Para poder eliminar la lista L se deben eliminar todos los nodos, por lo que toma n pasos y un paso extra para eliminar la referencia de la lista misma.
 - Complejidad es $O(n)$.