

Datos Lineales

22 OCTUBRE

Programación de Estructuras de Datos

Adrian Becerra Meza

Profesor: Luis Ricardo Peña Llamas

Grupo: 13

Listas Doblemente Ligadas

Una lista ligada es una colección ordenada de elementos del mismo tipo, las cuales están conectadas por punteros. Consiste en nodos donde cada nodo contiene campos de información y una referencia al siguiente nodo en la lista.

Para acceder a cualquier nodo en una lista ligada, tenemos que cursar secuencialmente la lista ligada completa hasta el elemento. En una lista ligada se pueden guardar nuevos elementos en cualquier lugar de la memoria.

En las listas ligadas existen:

- Lista individualmente ligadas
- Lista doblemente ligada
- Lista ligada circular
- Stack
- Cola
- Cola de prioridad

Lista Doblemente Ligada

Es un tipo de lista ligada donde cada elemento a parte de guardar información tiene dos ligas. La primera liga apunta al nodo anterior en la lista y la segunda liga apunta al siguiente nodo en la lista. El primer nodo de la lista tiene su liga anterior apuntando a null y de la misma manera el último nodo de la lista tiene su siguiente nodo apuntando a NULL.



Estas dos ligas nos permiten caminar por la lista ya sea de manera frontal o en reversa. Pero añadir una liga adicional requiere de espacio adicional.

Se usan las listas doblemente ligadas para guardar información lineal de tipos similares, pero los arreglos tienen ciertas limitantes.

- El tamaño de los arreglos están definidos
- Insertar un nuevo elemento en el arreglo de elementos es caro porque se debe crear espacio adicional para los nuevos elementos y ese espacio adicional sus elementos existentes tienen que cambiarse.
- Utilizan más memoria que los arreglos debido al almacenamiento utilizado por los punteros

Como en esta situación problema estamos trabajando con datos similares, entonces es buena idea utilizar listas doblemente ligadas, ya que es mas eficiente que otras listas ligadas.

Complejidad

Operación	Peor Caso	Caso Promedio
Insertar al inicio o al final	$O(1)$	$O(1)$
Borrar al inicio o al final	$O(1)$	$O(1)$
Buscar	$O(n)$	$O(n)$
Acceder	$O(n)$	$O(n)$

Reflexión

En lo particular yo creo que podríamos trabajar con otro tipo de listas ligadas, ya que su complejidad es menor, por lo que gastamos menos espacio en memoria y podemos de esta manera almacenar más datos. En nuestro caso utilizamos una fila priorizada, donde ingresamos las concurrencias obtenidas por medio de un hashmap, con esto pudimos lograr un proceso bastante veloz y eficaz.

Bibliografía

- Rahman, M. (s. f.). *PHP 7 Data Structures and Algorithms*. O'Reilly Online Learning.
Recuperado 22 de octubre de 2021, de <https://www.oreilly.com/library/view/php-7-data/9781786463890/c5319c42-c462-43a1-b33d-d683f3ef7e35.xhtml>
- uniwebsidad. (s. f.). *16.5. Otras listas enlazadas (Algoritmos de Programación con Python)*.
Recuperado 22 de octubre de 2021, de <https://uniwebsidad.com/libros/algoritmos-python/capitulo-16/otras-listas-enlazadas>

