

● Linked List에 대해서 설명하라

▶ Linked List는 노드(Node)라는 구조체로 이루어져 있는데, 노드는 데이터 값과 다음 노드의 주소(next address)를 저장합니다. Linked List는 물리적인 메모리상에서는 비연속적으로 데이터가 저장 되지만, Linked list를 구성하는 각각의 노드가 다음 노드의 주소를 가리킴으로써, 불연속적인 데이터를 연결하여 논리적인 연속성을 가진 자료구조입니다. 장점으로는 데이터가 추가 되는 시점에서 메모리를 할당하기 때문에 메모리를 좀 더 효율적으로 사용할 수 있고, 메모리상에서 연속성을 유지하지 않아도 되기 때문에 메모리 사용이 좀 더 자유롭습니다. 또한, 비순차적인 추가/삭제 시에도 다른 데이터들을 물리적으로 옮길 필요 없이 다음 노드 주소값만 변경하면 되기 때문에 $O(1)$ 의 시간 복잡도로 빠른 추가/삭제가 가능합니다. Linked List의 단점으로는 다음 노드의 주소를 추가적으로 저장해야 하기 때문에 데이터 하나당 차지하는 메모리가 더 크다는 것입니다. 또한 조회에 걸리는 시간이 $O(n)$ 으로 비교적 느립니다. 이는 Array와는 달리 Random access가 불가능하기 때문에, 순차적인 접근밖에 할 수 없기 때문입니다.

▶ Linked List는 tree, graph등 다른 자료구조를 구현할 때 자주 쓰이는 기본 자료구조입니다. 처음 노드는 head라는 포인터로 가리키고 있습니다. 마지막 노드의 next address는 null(0X0000)값으로 저장되어 있습니다.

▶ 시간 복잡도

Array의 경우 중간에 데이터를 추가/삭제하게 되면 해당 인덱스의 뒤에 있는 모든 원소들은 옮겨야 하다 보니 $O(n)$ 의 시간복잡도를 갖게 되었습니다. 하지만 Linked list를 물리적으로 옮길 필요없이 next address가 가리키는 주소값만 변경하면 되기 때문에 $O(1)$ 의 시간복잡도로 추가/삭제가 가능합니다.

	Array
추가/삭제 (insertion/deletion)	$O(1)$
조회 (access)	$O(n)$
탐색(search)	$O(n)$

Resources: inflearn 개발남 노씨