★ **Array와 Linked List의 가장 큰 차이점**은 **메모리에 저장되는 방식**과 **그에 따른 operation의 연산 속도**(time flexity)다.

● Array(배열)는 어떤 자료 구조인가?
▶Array는 **관련된 데이터를 메모리상에 연속적으로 미리 할당된 크기만큼 저장**하는 자료구조입니다.

● Array의 특징?
▶Array의 특징으로는 **고정된 저장공간**(fixed-size)와 **순차적인 데이터 저장**(order)이 있습니다.

● Array의 장점?
▶Array의 장점은 **조회(lookup)가 빠르다는 것으로, 조회를 자주 해야 하는 작업에서는 Array 자료 구조를 많이 씁니다**.

● **연산 속도 — 시간 복잡도** (time complexity)

	Array
조회(lookup/random access) • n번째 인덱스의 값에 접근하는 시간 • 배열의 주소 + 데이터 타입의 크기 x index => 인덱스가 n인 데이터의 주소 : 배열엔 연속적으로 데이터가 저장돼 있기 때문에 n번째 요소의 주소를 아주 빨리 계산해서 알아낼 수 있어서, 조회가 빠르다.	O(1) - Big O 1
마지막 인덱스에 추가(append)/**마지막 인덱스부터 삭제**(delete)	O(1)
insertion/deletion	O(n) 추가/삭제 후에도 데이터가 순차적으로 연결돼 있어야 하기 때문에, 데이터를 n번 옮겨야 하기 때문에 Big O n이 걸린다.
탐색(search) • 요소가 몇 번째 인덱스에 있는지 찾는 시간	O(n) 일일이 첫번째 인덱스부터 n번째 인덱스까지 확인해야 하기 때문에 Big O n의 시간 복잡도를 가진다.

● Array의 단점?
▶ Array의 단점은 **고정된 저장공간 (fixed-size) 특정상 선언 시에 크기를 미리 정해야 한다는 것인데, 이것 때문에 메모리 낭비나 추가적인 오버헤드**(만약 Array의 크기가 데이터 갯수보다 작을 경우에 더 큰 새로운 Array를 만들어서 데이터를 복사해오고 기존의 Array를 삭제하는 등의 추가적인 작업)**가 발생할 수 있습니다**.

● 미리 예상한 것보다 더 많은 수의 data를 저장하느라 Array의 size를 넘어서게 됐을 경우, 어떻게 해결할 수 있을까?

▶ 기존의 size보다 더 큰 Array를 선언을 해서 데이터를 옮겨 할당하고, 모든 데이터를 옮긴 후엔 기존 Array는 메모리에서 삭제하면 됩니다. 이런 식으로 동적으로 배열의 크기를 조절하는 자료구조를 Dynamic array라고 합니다.
또 다른 방법으로는, size를 예측하기 쉽지 않으면, Array대신 Linked list를 사용해서 데이터가 추가 될 때마다 메모리공간을 할당받는 방식을 사용하면 됩니다.

Resources: inflearn 개발남 노씨