

기술면접 - Array vs List (배열과 선형리스트 차이점)

Array(배열) VS List

- Array과 List의 가장 큰 차이점은 동적으로 크기 할당 여부이다.
- **Array는 동적으로 크기 할당이 불가능하고, List는 가능하다.**
: Array는 생성 시에 N개의 크기로 생성한다면 N개 이상의 크기로 늘릴 수 없다는 것을 말한다. 반면에 List는 add, remove 메서드를 통해서 크기를 자유롭게 늘리고, 줄일 수 있다.

배열의 특징

- 같은 자료형을 가진 변수를 하나로 나타낸 것
- 연속된 메모리 공간으로 이루어져 있는 것
- 정적 표현
- 인덱스를 이용하여 표현
- 지역성을 가지고 있다

배열의 장점

- 인덱스를 통한 검색이 용이함
- 연속적이므로 메모리 관리가 편하다

배열의 단점

- 한 데이터를 삭제하더라도 배열은 연속적이므로 공간이 남는다. 즉, 메모리 낭비가 있다
- 정적이므로 배열의 크기를 컴파일 이전에 정해주어야 한다.
- 컴파일 이후 배열의 크기를 변동할 수 없다.

배열의 시간 복잡도

- 삽입/삭제
배열의 맨 앞에 삽입/삭제하는 경우 : $O(n)$
배열의 맨 뒤에 삽입/삭제하는 경우 : $O(1)$
배열의 중간에 삽입/삭제하는 경우 : $O(n)$
- 탐색
 $O(1)$

배열은 언제 사용하나?

- 데이터 개수가 확실하게 정해져 있을 때
- 데이터의 삭제와 삽입이 적을 때
- 검색을 해야 할 때

리스트(List)의 특징

- 순서가 있는 데이터의 집합(데이터를 순차적으로 저장하는 선형 자료구조)
- 불연속적으로 메모리 공간을 차지
- 동적 표현 : 크기가 고정되어 있지 않음
- 인덱스가 없음 : 인덱스 접근이 불가능하다.
- 포인터를 통한 접근

리스트의 장점

- 포인터를 통하여 다음 데이터의 위치를 가리키고 있어 삽입 삭제가 용이하다.
- 동적이므로 크기가 정해져 있지 않다.
- 메모리의 재사용 편리
- 불연속적이므로 메모리 관리의 편리

리스트의 단점

- 검색 기능이 좋지 않다.
- 포인터를 통해 다음 데이터를 가리키므로 추가적인 메모리 공간 발생

리스트의 시간 복잡도

- 삽입/삭제
리스트의 맨 앞/뒤에 삽입/삭제하는 경우 : $O(1)$
리스트의 중간에 삽입/삭제하는 경우 : $O(n)$ (탐색하는 시간)
- 탐색
 $O(n)$

리스트는 언제 사용하나?

- 크기가 정해져 있지 않을 때
- 삽입과 삭제가 자주 일어날 때
- 검색을 자주 하지 않을 때

배열 / 리스트는 언제 사용해야 하는가?

- 가장 중요한 포인트는 해당 데이터를 할당 시, 데이터의 크기가 정해져 있는 것의 차이
- 정해져있다면 배열, 정해지지 않았다면 리스트를 사용하는 것이 가장 좋은 방법

list, array, arraylist의 차이?

Array(배열)	ArrayList	List
<ul style="list-style-type: none">- 고정된 배열 크기를 갖는다. (선언시 크기를 지정해주고, 삭제 및 추가와 같은 변형이 불가능하다.)- 같은 타입만 저장가능하다. (type safe하다)- 다차원 배열 입력이 가능하다.	<ul style="list-style-type: none">- 고정되지 않는, 추가/삭제의 변형이 가능한 객체타입이다.- 제네릭 타입으로서 서로 다른 타입의 데이터가 배열에 저장가능하다. 때문에 데이터를 가져올 때 박싱, 언박싱이 발생하며, type-safe하지 못한 이슈가 있다.	<ul style="list-style-type: none">- 마찬가지로 고정되지 않는 가변 객체 타입이다.- ArrayList의 단점을 보완하여 컴파일시 배열의 타입추론을 한다. 즉, 같은 타입만 저장가능하고, 때문에 박싱/언박싱이 발생하지 않는다.