

분류	질문	답변	중요도	관련 개념	출처
	좋은 코드의 조건이 될까요? 좋은 알고리즘의 조건이 될까요?				
	배열에 대해 설명해주세요	<p>선형 자료구조 - 랜덤 접근 가능 - 배열</p> <p>말 그대로 선형구조로 된 자료를 탐색하는 방법이다. 보통 어떤 값이 어디에 있는지 알아내는 게 목적이다.</p> <p>"O(1)"은 알고리즘의 시간 복잡도를 나타내는 표기법 중 하나로, 상수 시간을 의미 모든 자료에 O(1)으로 접근이 보장되는 자료구조들이다. "O(1)으로 접근 가능한 자료구조"란 어떤 연산이 입력 크기와 관계없이 <b>항상 일정한 시간 내에 실행되는 자료구조를 의미</b> 즉, <b>입력의 크기에 관계없이 항상 같은 시간만큼 소요</b>된다는 뜻 랜덤 접근 가능한 자료구조는 인덱스를 사용하여 직접 접근할 수 있기 때문에, 원하는 위치에 있는 데이터를 빠르게 찾을 수 있습니다.</p> <p>가장 쉬운 자료구조이자 언어에 따라 활용법이 가장 크게 달라지는 자료구조. 배열이란 연관된 데이터를 하나의 변수에 그룹핑해서 관리하기 위한 방법. 배열을 이용하면 하나의 변수에 여러 정보를 담을 수 있고, 반복문과 결합하면 많은 정보도 효율적으로 처리할 수 있습니다.</p> <p>배열의 단점입니다. 배열은 인덱스에 따라서 값을 유지하기 때문에 엘리먼트가 삭제돼도 빈자리가 남게 됩니다. *삭제된 자리를 뒤에 위치한 엘리먼트로 메꾸는 것입니다. 이렇게 데이터가 순서에 따라서 빈틈없이 연속적으로 위치하는 데이터 스트럭처를 리스트(list) <b>인덱스가 중요한 경우는 배열</b>을 사용하면 됩니다. null을 처리에서 제외해야 한다면 조건문을 사용하면 되겠죠. <b>인덱스가 중요하지 않은 경우에는 리스트를</b> 사용하면 됩니다.</p>	상		<a href="https://opentutorials.org/module/1335/8677">https://opentutorials.org/module/1335/8677</a>
	링크드 리스트에 대해 설명해주세요	<p>선형 자료구조 - 랜덤 접근 불가능 - 링크드 리스트</p> <p>말 그대로 선형구조로 된 자료를 탐색하는 방법이다. 보통 어떤 값이 어디에 있는지 알아내는 게 목적이다.</p> <p>모든 자료에 O(1)으로 접근이 보장되지 않는 자료구조들이다. 연결 리스트는 <b>각 노드가 다음 노드를 가리키는 방식으로 데이터를 저장하는 자료구조</b>입니다. 특정 인덱스에 있는 데이터에 접근하려면 <b>인덱스의 시작부터 순차적으로 노드를 탐색해야 하므로, 최악의 경우 O(n) 시간이 걸릴 수 있습니다</b> (여기서 n은 리스트의 길이). 랜덤 접근이 불가능한 자료구조들은 데이터에 접근할 때 순차적인 탐색이 필요하거나, 특정 위치에 바로 접근할 수 없어서 입력의 크기에 따라 접근 시간이 달라질 수 있습니다.</p> <p>값과 다음 노드를 가지고 있다. 옵션으로 이전 노드를 가지게 할 수도 있으며, 맨 뒤 노드가 맨 앞 노드를 다음 노드로 가지게 할 수도 있다. 또한, 중간에서 삽입과 삭제를 할 수 있다. 가장 간단하게 구현한 것은 위의 큐. 다만, 링크드 리스트는 원소들이 이곳 저곳에 흩어져있어서 구현체의 속도가 느리기 때문에, 잘 사용되지는 않는 편이다.</p>	상		
	배열과 링크드 리스트를 비교해주세요	<p>- 배열은 배열이 길이 만큼 <b>즉 한 상태에서 요소를 추가하면 배열의 길이의 두배에 해당하는 메모리를 미리 할당</b>하여 그 만큼 <b>불필요한 메모리를 사용</b>하게 됩니다. - 배열은 push와 pop을 제외한 모든 배열의 요소 추가 및 제거 메서드에서 O(n)의 시간 복잡도를 가집니다. 만약 <b>첫번째 인덱스에 요소를 추가해야 할 경우 나머지 모든 요소들을 오른쪽으로 한 칸씩 옮겨야</b>합니다.</p> <p>- 연결 리스트는 필요한 만큼만 메모리가 늘어나고 줄어들며 첫번째 인덱스에 요소를 추가하더라도 O(1)의 시간 복잡도를 가집니다. 따라서 또 다른 대표적인 선형 구조인 큐의 구현을 할 때 배열로 구현하는 것보다 연결 리스트를 활용하는 것이 훨씬 효율적입니다. - 또한 각 노드를 <b>차례대로 접근</b>해야하는 경우에도 배열보다 연결 리스트가 좋습니다.</p> <p><b>배열과 비교한 링크드 리스트의 특징</b> - 배열에 비해 공간 복잡도에서 이점을 가집니다. - 순서대로 데이터에 접근해야 할 때 매우 효율적입니다. - 첫번째 인덱스에 요소를 추가할 때 배열에 비해 시간 복잡도에서 이점을 가집니다. O(1) vs O(n) - 특정 인덱스에 랜덤하게 접근해야 하는 경우 배열을 사용하는 편이 효율적입니다. O(1) vs O(n)</p>	상		
	자바스크립트로 링크드 리스트 구현하는법	<p>연결 리스트의 요소(elements)들은 특정 메모리 주소나 인덱스에 저장되지 않습니다. 오히려 각 요소는 포인터 또는 다음 객체에 대한 링크를 가지는 독립적인 객체에 가깝습니다</p> <p><b>연결 리스트의 각 요소를 노드(node)</b>라 부릅니다. 노드는 일반적으로 데이터 그리고 <b>다음 노드를 가리키는 링크</b>, 이 2가지 아이템으로 구성됩니다 연결 리스트의 가장 <b>첫 번째 지점</b>을 <b>헤드(head)</b>라고 부릅니다. <b>헤드는 연결 리스트의 첫 번째 노드</b>를 의미합니다. <b>마지막 노드는 null</b>을 가릅니다. 만약 <b>연결 리스트가 비어있는 경우, 헤드는 null을 참조</b>하게 됩니다.</p> <p><b>자바스크립트로 연결 리스트를 표현</b> const list = {   head: {     value: 6     next: {       value: 10       next: {         value: 12         next: {           value: 3           next: null         }       }     }   } }; };</p>	상		<a href="https://www.freecodecamp.org/korean/news/implementing-a-linked-list-in-javascript/">https://www.freecodecamp.org/korean/news/implementing-a-linked-list-in-javascript/</a>

	링크드 리스트의 장점과 단점	<p><b>장점</b></p> <ul style="list-style-type: none"><li>- 데이터 구조의 큰 틀을 바꾸지 않고 노드를 추가하거나 삭제하기 쉽다는 장점 (배열과 대비되는 점)</li></ul> <p><b>단점</b></p> <ul style="list-style-type: none"><li>- 연결 리스트는 탐색이 느립니다. 배열과 달리, 연결 리스트는 데이터에 무작위 접근(random access)을 할 수 없기 때문입니다. 노드들은 첫 번째 노드부터 순차적으로만 접근해야 합니다.</li><li>- 연결 리스트는 배열보다 더 많은 메모리를 사용합니다. 왜냐하면 각 노드는 포인터를 담고 있기 때문입니다.</li></ul>	중		
	링크드 리스트의 유형	<p><b>3가지</b></p> <ul style="list-style-type: none"><li>- <b>단일 연결 리스트(Singly Linked Lists)</b> : 각 노드는 하나의 포인터만 가집니다. 우리가 위에서 이야기한 유형이 단일 연결 리스트입니다.</li><li>- <b>이중 연결 리스트(Doubly Linked Lists)</b> : 각 노드는 2개의 포인터를 가지는데, 하나는 다음 노드를 그리고 나머지 하나는 이전 노드를 가르킵니다.</li><li>- <b>원형 연결 리스트(Circular Linked Lists)</b> : 연결 리스트를 응용한 유형으로, 마지막 노드의 포인터가 첫 노드 또는 특정 노드를 가르키고 있는 마치 루프 형태를 가지는 유형을 말합니다.</li></ul>	중		
	단일 연결 리스트와 이중 연결 리스트를 비교 설명해주세요.	<p>연결 리스트는 앞뒤로 연결이 되었는지 아닌지 여부에 따라 단일(Singly) 연결 리스트와 이중(양방향, Doubly) 연결 리스트가 있다.</p> <p>단일 연결 리스트는 한 방향으로만 연결되어 있다. 따라서 나와 연결된 다음 노드는 알 수 있지만 내 이전 노드는 알 수 없다. 그러므로 역행해서 바로 이동할 수도 없다.</p> <p>반면 이중 연결 리스트는 이전과 이후 노드를 모두 저장하고 있어 양방향 이동이 자유롭다.</p> <p><b>단일 연결 리스트는 각 노드가 다음 노드만을 참조하지만 이중 연결 리스트에서는 이전 노드와 다음 노드 양 쪽을 참조합니다.</b></p>	중		<a href="https://velog.io/@717lumos/%EC%9E%90%EB%A3%8C%EA%B5%AC%EC%A1%B0-%EC%97%B0%EA%B2%B0%EB%A6%AC%EC%8A%A4%ED%8A%B8Linked-List-%EB%8B%A8%EC%9D%BC%EC%97%B0%EA%B2%B0%EB%A6%AC%EC%8A%A4%ED%8A%B8-%EC%9D%B4%EC%A4%91%EC%97%B0%EA%B2%B0%EB%A6%AC%EC%8A%A4%ED%8A%B8">https://velog.io/@717lumos/%EC%9E%90%EB%A3%8C%EA%B5%AC%EC%A1%B0-%EC%97%B0%EA%B2%B0%EB%A6%AC%EC%8A%A4%ED%8A%B8Linked-List-%EB%8B%A8%EC%9D%BC%EC%97%B0%EA%B2%B0%EB%A6%AC%EC%8A%A4%ED%8A%B8-%EC%9D%B4%EC%A4%91%EC%97%B0%EA%B2%B0%EB%A6%AC%EC%8A%A4%ED%8A%B8</a>
	<a href="https://www.freecodecamp.org/korean/news/implementing-a-linked-list-in-javascript/">https://www.freecodecamp.org/korean/news/implementing-a-linked-list-in-javascript/</a>				
	<a href="https://prgms.tistory.com/134">https://prgms.tistory.com/134</a>				