

공간복잡도

📚 공간 복잡도란?

공간 복잡도(Space Complexity)는 알고리즘이 실행될 때 메모리(공간)를 얼마나 많이 사용하는지를 나타냅니다.

즉, 알고리즘이 작업을 수행하면서 필요한 추가 메모리 공간의 크기를 정량화하는 지표입니다.

*수행 능력에 관한건 [시간복잡도](#) 입니다.

🔍 핵심 개념 정리

항목	설명
정의	알고리즘이 실행 중에 사용하는 메모리의 양 (예: 변수, 배열, 스택 등)
표기법	일반적으로 $O(f(n))$ 형태로 표현 (예: $O(1)$, $O(n)$, $O(n^2)$)
예시	단순한 변수는 $O(1)$, 배열을 저장할 경우 $O(n)$ 등

✓ 예시 설명

알고리즘	공간 복잡도	설명
선형 검색	$O(1)$	변수만 사용하므로 상수 공간 필요
재귀 함수 (단순한)	$O(n)$	재귀 스택이 깊이 증가하면 스택 공간 증가
배열 복사	$O(n)$	입력 배열을 복사하므로 추가 배열 필요
정렬 알고리즘 (병합 정렬)	$O(n)$	추가 배열을 위해 공간 필요

⚠ 주의할 점

- 시간 복잡도와는 다르게, 공간 복잡도는 시간보다는 메모리 사용량에 초점을 맞춥니다.
- 실제 프로그래밍에서는 메모리 제한(예: 제한된 스택, RAM)이 있어, 효율적인 공간 사용이 중요합니다.
- 예: 큰 데이터를 처리할 때, 공간 효율적인 알고리즘(예: 원소를 하나씩 처리하는 방식)이 더 유리할 수 있음.

💡 요약

공간 복잡도는 알고리즘이 메모리 사용량 측면에서 얼마나 효율적인지를 나타냅니다.

- $O(1)$: 상수 공간 (예: 변수만 사용)
- $O(n)$: 선형 공간 (예: 배열 복사, 재귀 깊이)
- $O(n^2)$: 제곱 공간 (예: 2차원 배열 생성 등)

이 정보는 알고리즘 설계, 시스템 최적화, 프로그래밍 언어 선택 등에 중요한 역할을 합니다.

필요하면 예제 코드나 더 깊이 있는 분석도 도와드릴 수 있어요! 😊



팁: 알고리즘을 평가할 때는 시간 복잡도와 공간 복잡도를 함께 고려하세요.

예: 퀵소트는 평균 시간 복잡도가 $O(n \log n)$ 이지만, 최악의 경우 $O(n^2)$ 이며, 공간 복잡도는 $O(\log n)$ (재귀 스택)입니다.

필요하면 이 내용을 Obsidian 노트에 정리해드릴 수도 있어요! 📝