

<용어체크>

분할 정복

주어진 문제의 입력을 다루기 쉽게 부분으로 분할하여 문제를 해결(정복)하는 방식이며 문제를 더 이상 나눌 수 없을 때까지 나누고 이렇게 나뉘어진 문제들을 각각 풀으로써 전체 문제의 답을 얻는 알고리즘이다.

분할

분할은 주어진 문제를 여러 개의 부분 문제들로 나누는 것을 의미하며 문제가 작아지면 작아질수록 풀기 쉬워지는 성질을 이용한 것이다.

정복

정복이란 분할된 부분 문제들의 부분해를 찾는 것을 의미하며 정복하는 방법은 문제에 따라 다르나 일반적으로 부분 문제들의 해를 취합하여 보다 큰 부분 문제의 해를 구한다.

<학습내용>

분할 정복

분할 정복의 적용

분할 정복의 장단점

<학습목표>

분할 정복의 특징에 대해 설명할 수 있다.

분할과 정복으로 문제를 나눠서 해결하는 방법에 대해 설명할 수 있다.

분할 정복의 장점과 단점에 대해 설명할 수 있다.

Q. 해결해야 할 문제의 규모가 큰 경우 좀 더 작은 여러 개의 문제로 분할하고 각 문제를 해결한 다음 결과를 합쳐 원래의 문제를 해결하는 방법이 있는데 이러한 방법이 항상 효율적일까요?

: 규모가 큰 문제를 작은 규모의 모듈로 분할하여 문제를 해결하는 방법은 매우 유용하나 모듈로 나누면 모듈끼리 서로 통신하는 방법이 필요합니다. 또한 경우에 따라서는 문제를 분할함으로써 문제 자체는 작아졌지만 분할 후 입력의 크기가 몇 배로 늘어날 수도 있습니다. 이런 경우, 문제를 분할하고 하나로 취합하는 방법을 사용하는 것은 매우 부적절합니다. 또한 무작정 작게만 쪼개면 통신으로 인해 복잡도가 오히려 증가될 수 있으므로 설계자는 어느 수준으로 쪼갤지를 결정해야 합니다. 따라서 분할하고 취합하는 방법이 모든 경우에 효율적인 것은 아니며 해당 문제가 이러한 해결 방법에 잘 부합되어야 합니다.

분할 정복

주어진 문제의 입력을 다루기 쉽게 부분으로 분할하여 문제를 해결(정복)하는 방식의 알고리즘이다.

문제를 더 이상 나눌 수 없을 때까지 나누고 이렇게 나누어진 문제들을 각각 풀으로써 전체 문제의 답을 얻는 알고리즘이다.

문제를 두 단계인 분할과 정복으로 나눠서 해결하는 것이다.

분할한 입력에 대하여 동일한 알고리즘을 적용하여 해를 계산하며 이들의 해를 취합하여 원래 문제의 해를 얻는다.

분할 정복의 적용

병합 정렬은 입력이 2개의 부분 문제로 분할하여 부분 문제의 크기가 $1/2$ 로 감소하는 분할 정복 알고리즘이다.

병합 정렬은 외부 정렬의 기본이 되는 정렬 알고리즘이다.

연결 리스트에 있는 데이터를 정렬할 때에도 퀵 정렬이나 힙 정렬보다 훨씬 효율적이다.

멀티코어 CPU와 다수의 프로세서로 구성된 그래픽 처리 장치의 등장으로 정렬 알고리즘을 병렬화하는 데에 병합 정렬 알고리즘이 활용된다.

분할 정복의 장단점

분할 정복의 장점

- ✓ 문제를 나눔으로써 어려운 문제를 해결할 수 있다.
- ✓ 문제를 나누어 해결한다는 특징상 병렬적으로 문제 해결이 가능하다.

분할 정복의 단점

- ✓ 함수를 재귀적으로 호출함으로 인해 함수 호출로 인한 오버헤드가 발생한다.
- ✓ 스택에 다양한 데이터를 보관하고 있어야 하므로 스택 오버플로우가 발생하거나 과도한 메모리 사용을 하게 될 수도 있다.