

이. 자료구조와 알고리즘

1.1 자료구조와 알고리즘

프로그래밍 = 자료구조 + 알고리즘
자료 처리

알고리즘 (algorithm) : 컴퓨터로 문제를 풀기 위한 단계적인 절차

문제의 컴퓨터가 풀 수 있는 상황에서 문제를 해결하는 방법은 정답에 도달할 수 있는 여러 가지가 있을 수 있다

특정한 일을 수행하는 방법의 집합

입력 : 0개 이상의 입력이 존재해야 한다

출력 : 1개 이상의 출력이 존재해야 한다

명확성 : 각 명령어의 의미를 분명히 알고 명확해야 한다

유한성 : 한정된 수의 단계 후에 반드시 종료되어야 한다

효율성 : 각 명령어들을 좀더 간단, 또는 컴퓨터로 실행 가능한 언어이어야 한다

알고리즘 기술 방법

① 순서, 영미어 자연어

② 흐름도 (Flowchart)

③ 의사코드 (Pseudo-code)

④ 프로그래밍 언어

1.2 추상 자료형

추상 자료형 (ADT: abstract data type) : ADT의 이름, 객체 (objects), 함수 (functions)

1.3 알고리즘의 성능 분석

알고리즘 복잡도 분석 (complexity analysis) < 시간 복잡도 (time complexity) : 수행 시간
공간 복잡도 (space complexity) : 알고리즘이 차지하는 공간

빅 오 표기법 $O(n)$: 가장 큰 차수에 영향을 받는 표기법 (상한)

$O(1) > O(\log n) > O(n) > O(n \log n) > O(n^2) > O(n^3) > O(2^n) > O(n!)$

→ 정확히 각속도 시간이 증가됨

빅 오메가 표기법 $\Omega(n)$: 가장 작은 차수에 영향을 받는 표기법 (하한)

빅세타 표기법 $\Theta(n)$: 가장 큰 차수와 작은 차수에 영향을 받는 표기법 (상·하한)

최선의 경우 (best case) : 수행 시간 가장 빠름, 예외 X

평균의 경우 (average case) : 수행 시간 평균 계산 방법

최악의 경우 (worst case) : 수행 시간 가장 느림, 널리 분포