계산복잡도

Computational Complexity

크리스토스 H. 파파디미트리우
Chistos H. Papadimitiriou
캘리포니아 대학교 샌디에이고

번역 김태원
2023년 8월 8일

차 례

제 I 편

알고리즘

알고리즘 책은 계산복잡도를 다루는 장으로 끝나기 마련이니, 알고리즘에 관한 기본 사실 몇 가지 돌이키며 본고를 시작하는 편이 적절하겠다. 이어지는 세 장에서 우리의 목표는 간단하지만 중요한 요점을 조금 지적하는 것이다. 계산computational 문제란 해결되어야 하는 것일 뿐만 아니라, 탐구할 가치가 있는 객체이기도 하다는 점이다. 문제와 알고리즘은 수학적으로 형식화되고 분석될 수 있다. 차례로 이를테면 언어languages나 튜링장치Turing machines가 그렇다. 그리고 정확한 형식주의는 그닥 중요하지 않다. 다항시간 계산가능성Polynomial-time computability은 계산 문제에 대해 원하는 성질로, 실용적인 해결가능성practical solvability의 직관적인 개념과 동질이다. 여러 상이한 계산 모델models은 효율성의 다항 손실polynomial loss로 또 다른 모델을 시뮬레이션할 수 있다—비결정론nondeterminism이라는 예외, 즉 제 시뮬레이션에 지수시간exponential time을 요구하는 것으로 보이는 예외를 제외하면 말이다. 그리고 알고리즘을 아예 지니지 않는 문제가 존재하는데, 아무리 비효율적인 것조차 지니지 않는다.

제 1 장

문제와 알고리즘

알고리즘은 문제를 풀기 위한 자세한 스텝별step-by-step 방법론이다. 다만 문제problem가 뭔가? 우리는 이 장에서 중요한 예시 세 개를 소개한다

1.1 그래프 도달가능성

그래프 $G = (V, E)$는 노드nodes V와 선분edges E, 즉 노드 쌍의 집합이다 (이를테면 그림 1.1을 보라. 우리의 그래프는 모두 유한하고 유향directed이겠다). 여러 계산 문제는 그래프를 다룬다. 그래프에 관해 가장 기본적인 문제는 이것이다. 그래프 G와 두 노드 $1, n \in V$가 주어질 때 1에서 n까지 경로path는 존재하는가? 우리는 이 문제를 도달가능성REACHABILITY[†]이라고 부른다. 가령, 그림 1.1에는 분명 노드 1에서 $n = 5$까지 경로, 즉 $(1, 4, 3, 5)$가 존재한다. 만약 이 대신 선분 $(4, 3)$의 방향을 역전하면, 그런 경로는 존재하지 않는다.

대다수의 흥미로운 문제와 마찬가지로, 도달가능성은 가능한 *일례instances*의 무한 집합을 지닌다. 각 일례는 수학적인 객체로, (우리의 경우, 그래프와 그 두 노드로) 곧 우리가 질문을 묻고 답을 기대하는 대상이다. 이때 질문이 속한 특정 종류가 문제를 특징짓는다. 도달가능성은 "네" 혹은 "아니오" 가운데 하나의 답안을 요구하는 질문이라는 점에 유의하라. 이런 문제는 *결정문제decision problems*라고 부른다. 복잡도 이론에서 보통 우리는 온갖 상이한 답안을 요구하는 문제보단 결정문제만 다루는 편이 편리하게 통합적이고 단순하다고 본다. 그러니 결정문제는 본고에서 중요한 역할을 맡겠다.

[†]복잡도 이론에서, 계산 문제는 단지 풀어야 하는 것일 뿐 아니라, 다만 그 자체로 흥미로운 수학적 객체이기도 하다. 문제가 수학적 객체로 다뤄질 때, 그 이름을 대문자로 표기하겠다. [역자: 대문자 표기를 드러냄표(`circemph`)로 대체한다.]

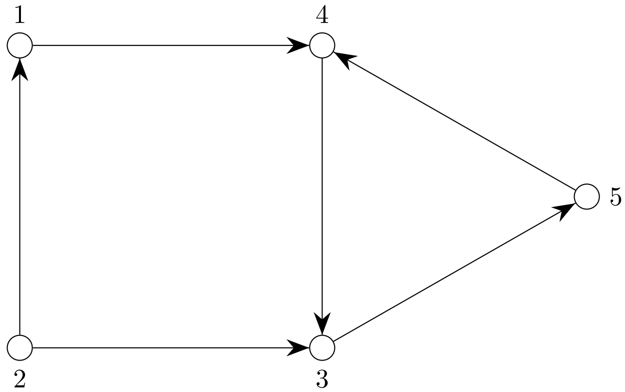

그림 1.1: 그래프.

gg