

## 알고리즘

**알고리즘**(영어: algorithm), **셈법**은 수학과 컴퓨터과학, 언어학 또는 연인 분야에서 어떠한 문제를 해결하기 위해 정해진 일련의 절차이다. 계산을 실행하기 위한 단계적 절차를 의미하기도 한다. 즉, 문제 풀이에 필요한 계산절차 또는 처리과정의 순서를 뜻한다. 프로그래밍명령어의 집합을 의미하기도 한다.

알고리즘은 연산, 데이터 마이닝(기계 학습) 또는 자동화된 추론을 수행한다.

### 이름

산법(算法), 셈법, 계산절차(計算節次)라고도 한다.

알고리즘은 9세기 페르시아의 수학자인 무함마드 알과리즈미의 이름을 라틴어화한 알고리스무스(Algorismus)에서 유래한 표현이다.

영어로 Algorithm의 발음 기호는 /ˈælɡəˌɪðəm/ 또는 /ˈælɡəˌɪðɪm/이며 Algorithm을 '알고리즘'으로 표기하는 것은 This를 '지스'로, Rhythm /rɪðəm/을 '리즘'으로 표기하는 것과 마찬가지로의 잘못된 것이라는 주장이 있다. 하지만 실제 생활에서는 알고리즘이라는 표기가 알고리즘이라는 표기에 비해 압도적으로 많이 사용되고 있다.<sup>[1]</sup>

### 정의

 [en:Algorithm characterizations](#) 문서를 참고하십시오.

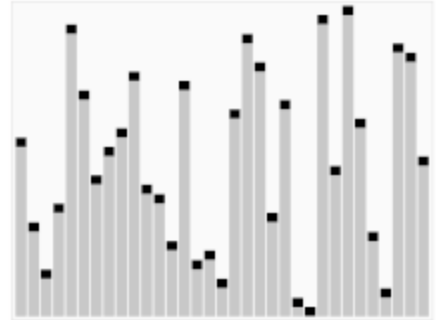
멈춤문제의 결과로 알고리즘멈추기까지 걸리는 시간을 일반적으로 측정할 수 없다.

그러므로 알고리즘에 대해 단순한 직관을 만족하는 형식적인 정의는 불가능하다.

따라서 알고리즘의 공식적인 정의는 없다.

대부분의 알고리즘은 유한한 수의 규칙에 따라 구별 가능한 기호들을 조작하여 입력 정수에서 출력 정수를 생성하기 위한 일반화된 작업을 정의한다. 다음은 좋은 알고리즘의 특징이다.

- **정밀성** : 변하지 않는 명확한 작업 단계를 가져야 한다.
- **유일성** : 각 단계마다 명확한 다음 단계를 가져야 한다.
- **타당성** : 구현할 수 있고 실용적이어야 한다.
- **입력** : 정의된 입력을 받아들일 수 있어야 한다.
- **출력** : 답으로 출력을 내보낼 수 있어야 한다.
- **유한성** : 특정 수의 작업 이후에 정지해야 한다.



퀵 정렬 알고리즘

- 일반성 : 정의된 입력들에 일반적으로 적용할 수 있어야 한다.

## 구현

알고리즘은 자연어, 의사코드, 순서도, 프로그래밍언어, 인터프리터가 작업하는 제어테이블, 유한상태 기계의 상태도 등으로 표현할 수 있다. 다음은 알고리즘 개발의 전형적인 단계이다.


문제 정의 → 모델 고안 → 명세 작성 → 설계 → 검증 → 분석 (복잡도 등) → 구현 → 테스트 → 문서화

알고리즘을 설계하는 기술에는 운용과학의 방법, 설계짜임새를 써먹는 방법 등이 있다. 대부분의 알고리즘은 컴퓨터 프로그램으로 구현되지만, 전기 회로나 생물학적 신경회로를 사용하기도 한다.

## 분류

- 구현 : 재귀적 알고리즘, 연역적 알고리즘, 결정론적 알고리즘, 근사 알고리즘, 양자 알고리즘 등.
- 설계 : 무차별 대입 공격, 분할 정복 알고리즘, 그래프 순회, 분기 한정법, 확률적 알고리즘, 리덕션, 백트래킹 등.
- 최적화 문제 : 선형 계획법, 동적 계획법, 탐욕 알고리즘, 휴리스틱 함수 등.
- 이론적 분야 : 검색 알고리즘, 정렬 알고리즘, 수치 알고리즘, 그래프 알고리즘, 문자열 알고리즘, 암호학적 알고리즘, 기계 학습, 데이터 압축 등.

## 복잡성


 이 부분의 본문은 시간 복잡도입니다.

입력의 크기가  $n$ 일 경우, 점근 표기법  $O$ 를 사용하여 다음과 같이 나타낸다.

- $O(1)$  :  $n$ 에 관계없이 일정 시간 이하에 수행되는 알고리즘이다. 예) 파일의 첫 번째 바이트가 널 (null)인지 검사하는 것.
- $O(\log n)$  :  $\log_2 n$ 에 비례하는 시간 이하에 수행되는 알고리즘이다. 예) 이진 탐색.
- $O(n)$  :  $n$ 에 비례하는 시간 이하에 수행되는 알고리즘이다. 예) 기수 정렬.
- $O(n \log n)$  :  $n$ 에 대략 비례할 수 있는 시간 이하에 수행되는 알고리즘이다. 예) 정렬 알고리즘.
- $O(n^2)$  :  $n^2$ 에 비례하는 시간 이하에 수행되는 알고리즘이다. 예) 최장 공통 부분 수열 문제.
- $O(n^3)$  :  $n^3$ 에 비례하는 시간 이하에 수행되는 알고리즘이다. 예) 행렬 곱셈.
- $O(a^n)$  :  $2^n$ 과 같은 꼴의 수행 시간 이하에 수행되는 알고리즘이다. 예) 충족 가능성 문제.
- $O(n!)$  :  $n!$  즉  $n \times (n-1) \times (n-2) \times \dots \times 1$ 과 같은 꼴의 수행 시간 이하에 수행되는 알고리즘이다. 예) 배열의 모든 순열을 검사하는 것.

대문자 O 표기법의 정의상 아래의 복잡도는 그 위의 복잡도를 포함하므로, 대부분의 알고리즘은  $O(n!)$ 의 수행 시간을 가진다.

## 예

 이 부분의 본문은 [알고리즘의 목록](#)입니다.

## 알고리즘의 예시

가장 단순한 알고리즘 가운데 하나는 임의로 나열된 숫자들 가운데 가장 큰 수를 찾는 것이다. 다음은 목록 안에 있는 모든 수를 살펴보는 알고리즘이다.

알고리즘 LargestNumber

Input: A list of numbers  $L$ .

Output: The largest number in the list  $L$ .

```
if  $L.size = 0$  return null
largest  $\leftarrow L[0]$ 
for each item in  $L$ , do
  if item > largest, then
    largest  $\leftarrow$  item
return largest
```

- " $\leftarrow$ "은 대입을 가리킨다. 이를테면 " $\alpha \leftarrow \beta$ "는  $\alpha$ 에  $\beta$ 를 대입하는 것을 뜻한다.
- "return"은 알고리즘을 종료하고 다음의 값을 출력한다.

## 같이 보기

- [Garbage in, garbage out](#)
- [Introduction to Algorithms](#)
- [계산 이론](#)
- [고속 푸리에 변환](#)
- [라메의 정리](#)
- [람다 대수](#)
- [순서도](#)
- [알고리즘 트레이딩](#)
- [처치-튜링 명제](#)
- [추상 기계](#)
- [휴리스틱 이론](#)
- [기사의 여행](#)

- 스테인하우스-존슨-트로터 알고리즘

## 각주

---

1. 구글에서 "알고리즘" 검색 결과 ([https://www.google.com/search?source=hp&ei=y1ksW9W3GcGC8wXj8aaYAw&q=%22%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98%22&oq=%22%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98%22&gs\\_l=psy-ab.3..0l10.1580.4304.0.4558.15.14.0.0.0.0.144.1560.3j11.14.0....0...1.1j4.64.psy-ab..4.11.1282.0..35i39k1j0i131k1j0i3k1.0.FNz0SVauo54](https://www.google.com/search?source=hp&ei=y1ksW9W3GcGC8wXj8aaYAw&q=%22%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98%22&oq=%22%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98%22&gs_l=psy-ab.3..0l10.1580.4304.0.4558.15.14.0.0.0.0.144.1560.3j11.14.0....0...1.1j4.64.psy-ab..4.11.1282.0..35i39k1j0i131k1j0i3k1.0.FNz0SVauo54)) 및 "알고리즘" 검색 결과 ([https://www.google.com/search?ei=0FksW7HcH5eA-QbEnZuwBg&q=%22%EC%95%8C%EA%B3%A0%EB%A6%AC%EB%93%AC%22&oq=%22%EC%95%8C%EA%B3%A0%EB%A6%AC%EB%93%AC%22&gs\\_l=psy-ab.3..0j0i10i30k1l9.4763.5877.0.6155.4.4.0.0.0.0.127.456.1j3.4.0....0...1.1j4.64.psy-ab..0.2.229...0i13k1.0.SeBb2k6bm3c](https://www.google.com/search?ei=0FksW7HcH5eA-QbEnZuwBg&q=%22%EC%95%8C%EA%B3%A0%EB%A6%AC%EB%93%AC%22&oq=%22%EC%95%8C%EA%B3%A0%EB%A6%AC%EB%93%AC%22&gs_l=psy-ab.3..0j0i10i30k1l9.4763.5877.0.6155.4.4.0.0.0.0.127.456.1j3.4.0....0...1.1j4.64.psy-ab..0.2.229...0i13k1.0.SeBb2k6bm3c))

## 외부 링크

---

- (영어) 알고리즘 (<https://curlie.org/Computers/Algorithms/>) - Curlie

원본 주소 "<https://ko.wikipedia.org/w/index.php?title=알고리즘&oldid=33036208>"