

알고리즘의 복잡도

(Complexity of Algorithms)

시간 복잡도(Time Complexity)

가장 긴 시간을 소요하게 만드는 입력에 따라 소요되는 시간

평균 시간 복잡도(Average Time Complexity)

임의의 입력 패턴을 가정했을 때 소요되는 시간의 평균

최악 시간 복잡도(Worst-case Time Complexity)

문제의 크기와 이를 해결하는 데 걸리는 시간 사이의 관계

Big-O Notation

- 점근 표기법 (asymptotic notation)의 하나
- 어떤 함수의 증가 양상을 다른 함수와의 비교로 표현
(알고리즘의 복잡도를 표현할 때 흔히 쓰임)
- $O(1)$, $O(\log n)$, $O(n)$, $O(n^2)$, $O(2^n)$ 등으로 표기

$O(1)$ - 상수 시간(Constant time)

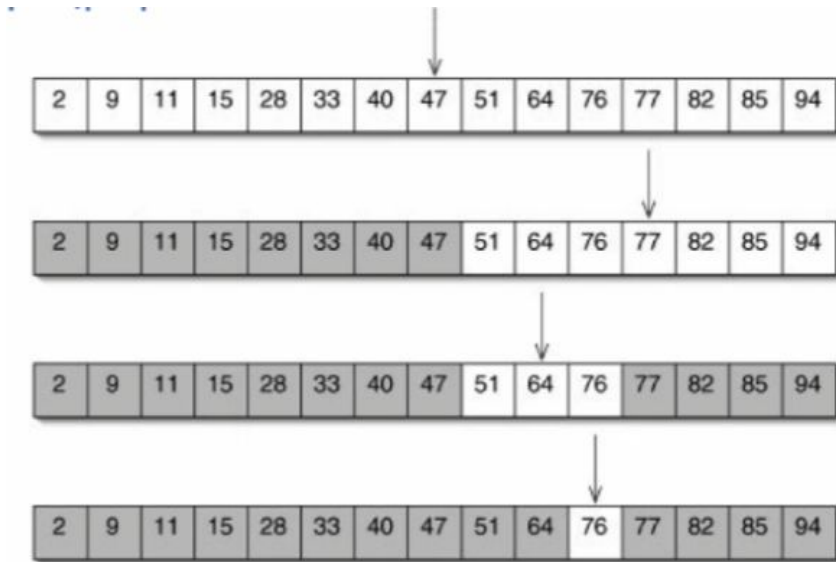
입력 데이터의 크기에 상관없이 언제나 일정한 시간이 걸림
(데이터가 얼마나 증가하든 성능에 영향을 거의 미치지~~x~~)

로그 시간 알고리즘 - $O(\log n)$

예: n 개의 크기 순으로 정렬된 수에서

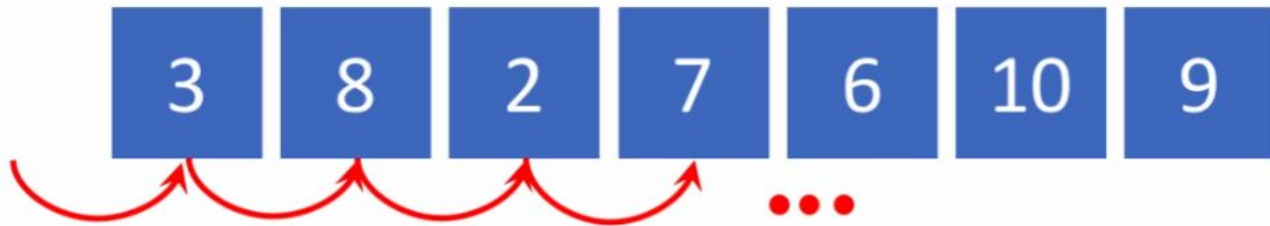
특정 값을 찾기 위해 이진 탐색 알고리즘을 적용

(이진트리)



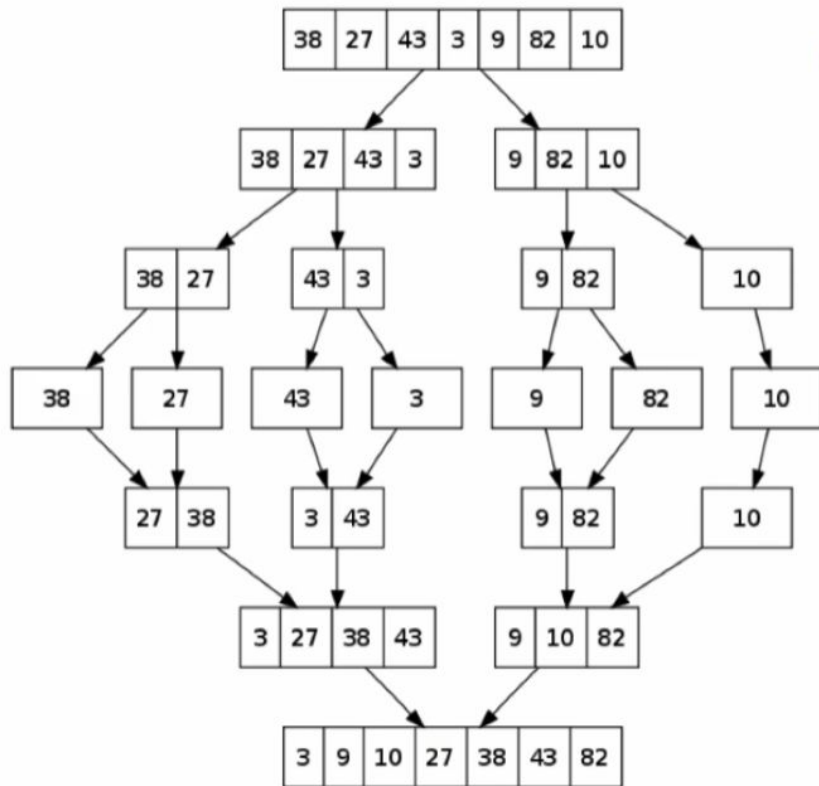
선형 시간 알고리즘 - $O(n)$

예: n 개의 무작위로 나열된 수에서 최댓값을 찾기 위해 선형 탐색 알고리즘을 적용 (for 문)



최댓값 - 끝까지 다 살펴 보기 전까지는 알 수 없음

선형 로그 시간 – $O(n \log n)$



예: 병합 정렬(merge sort)

이차 시간 알고리즘 - $O(n^2)$

54	26	93	17	77	31	44	55	20	Assume 54 is a sorted list of 1 item
26	54	93	17	77	31	44	55	20	inserted 26
26	54	93	17	77	31	44	55	20	inserted 93
17	26	54	93	77	31	44	55	20	inserted 17
17	26	54	77	93	31	44	55	20	inserted 77
17	26	31	54	77	93	44	55	20	inserted 31
17	26	31	44	54	77	93	55	20	inserted 44
17	26	31	44	54	55	77	93	20	inserted 55
17	20	26	31	44	54	55	77	93	inserted 20

예: 삽입 정렬

(insertion sort)

지수시간 알고리즘 - $O(2^n)$

데이터량이 많아질수록 처리시간이 기하급수적으로 늘어나는 알고리즘

예: 피보나치 수열, 재귀가 역기능을 할 경우

Big-O: functions ranking

BETTER



WORSE

- $O(1)$ constant time
- $O(\log n)$ log time
- $O(n)$ linear time
- $O(n \log n)$ log linear time
- $O(n^2)$ quadratic time
- $O(n^3)$ cubic time
- $O(2^n)$ exponential time

공간 복잡도(Space Complexity)

문제의 크기와 이를 해결하는 데 필요한 메모리 공간 사이의 관계

- 시간과 공간은 반비례적인 경향이 있음
- 최근 대용량 시스템이 보편화되면서 공간 복잡도보다는 시간 복잡도가 우선
- 알고리즘은 **시간 복잡도**가 중심

감사합니다.