2강: 알고리즘 성능 평가

노트북: [algorithm]

만든 날짜: 2020-10-13 오전 8:30 **업데이트**:

작성자: 황인범

2강 : 알고리즘 성

2020-10-13 오전 8:39

능 평가

복잡도란?

복잡도: 알고리즘 성능을 나타내는 척도

1. 시간 복잡도 - 수행시간 분석

2. 공간 복잡도 - 메모리 사용량 분석

복잡도와 성능 반비례

빅오 표기법(Big-O Notation)

가장 빠르게 증가하는 항만을 고려하는 표기

N3(제곱) + 5N(제곱) + 10000 이 있을때

가장 차수가 끈 N세제곱만 남겨 O(N3(제곱))

좋음(Better)	순위	명칭
	O(1)	상수 시간(Constant time)
	O(logN)	로그 시간(Log time)
	O(N)	선형 시간
	O(NlogN)	로그 선형 시간
	$O(N^2)$	이차시간
	$O(N^3)$	삼차시간
나쁨(Worse)	O(2")	지수 시간

간단한 시간 복잡도 계산

```
array = [1,2,3,4,5]

for x in arry:
    summary +=x

= > 시간 복잡도 O(N)

#2중 반복문

for i in arry:
    for j in array:
        temp = i * j
        print(temp)

=> 시간 복잡도 O(N제곱)
```

알고리즘 설계 Tip

코딩 테스트 문제에서 시간제한은 통상 1~5초 가량(시간제한이 있다면 1초 없다면 5초이내)

문제에서 가장 먼저 확인해야 하는 내용은 시간제한이다.

N의 범위 - 500 => O(N세제곱) 인 알고리즘 까지 가능

N의 범위 - 2,000 => O(N제곱) 인 알고리즘 까지 가능

N의 범위 - 100,000 => O(NlogN) 인 알고리즘 까지 가능

N의 범위 - 10,000,000 => O(N) 인 알고리즘 까지 가능

알고리즘 문제 해결 과정

- 1. 지문읽기 및 컴퓨터적 사고
- 2. 요구사항(복잡도) 분석
- 3. 문제 해결을 위한 아이디어 찾기
- 4. 소스코드 설계 및 코딩

핵심 아이디어를 캐치하여 간결한게 소스코드 형태로 제출하면 된다!!

시간을 측정하기 위한 과정

import time start_time = time.time() # 측정시작

프로그램 소스코드

end_time = time.time() #측정 종료

print("time:", end_time - start_time) # 수행 시간 출력