

OT & 시간복잡도

2025.01.24

23기 분석 김무연

목 차

1. OT

소개

진행방식

발제자/리뷰자 선정

2. 공부 방향성

공부 방향성 제안

3. 시간복잡도

복잡도란?

빅오 표기법

4. 계획

진행계획

목표

5. Q&A

OT

OT-스터디장 및 스터디원 소개

24기 분석 김무연

1

- 국민대학교 AI빅데이터융합경영학과 / 데이터사이언스 연계융합전공
- 관심분야: Computer Vision

2

다른 분들도 차례로 자기소개 한번씩 부탁드립니다!

OT - 진행방식

- 1 활동은 매주 금요일 22시 ~ 23시 비대면 (더 일찍 끝나거나 늦게 끝날 수도 있음)
- 2 매주 발제자를 선정하여 다음의 이론들에 대해서 30분 내외로 발제할 예정
 - 발제 자료는 이론반-발제자료 폴더에 정리 (깃헙)
- 3 발제자 외 팀원들은 발제자가 발제할 부분에 대한 예습(이코테 교재 등)을 진행하고 발제자가 선정한 연습문제가 있다면 풀어서 깃헙에 push/PR해야함
- 4 매주 발제한 내용에 대한 복습은 자율적으로 진행
 - 복습은 자율적으로 진행하되, 발제시간에 앞서 저번 내용에 관한 복습 문제 풀이에 대해서 한주에 한 분씩 발표하고 문제에 대한 QA/토론 예정

발제자 선정

1주차	OT,시간복잡도 (1/24)	김무연
2주차	그리디 (1/31)	김무연
3주차	DFS/BFS (2/7)	
4주차	정렬(바뀔 수도 있음, 2/14)	
5주차	이진탐색 (2/21)	
6주차	다이내믹 프로그래밍 (2/28)	
7주차	최단경로 (3/7)	

리뷰자 선정

1주차	OT,시간복잡도 (1/24)	김무연
2주차	그리디 (1/31)	
3주차	DFS/BFS (2/7)	
4주차	정렬(바뀔 수도 있음, 2/14)	
5주차	이진탐색 (2/21)	
6주차	다이내믹 프로그래밍 (2/28)	
7주차	최단경로 (3/7)	

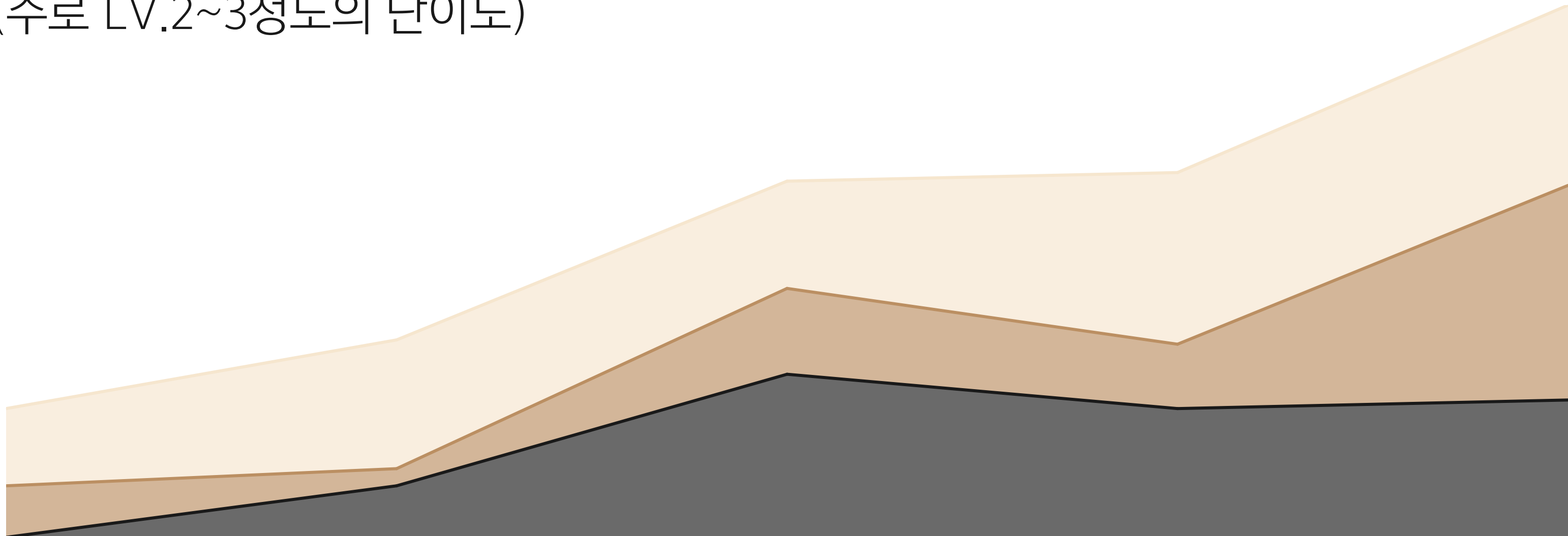
공부방향성 제안

from BABO 4th 스터디장
22기 분석 원종빈

문제 풀이 사이트 추천: 백준 + solved.ac 에서 차례로 올라가며 푸는 것을 추천!

- 추가로 프로그래머스, Leetcode 도 있음

프로그래머스 <- 실제 코테에서 자주 사용된다고 합니다!
(주로 LV.2~3정도의 난이도)



시간 복잡도

What is complexity?

알고리즘의 성능 분석

- 알고리즘의 실행 시간 분석
 - 방법: time 함수를 사용하여 start의 시간과 end의 시간의 차를 구한다.
 - 단점: SW/HW에 따라 다를 수 있음

시간복잡도

특정한 크기의 입력에 대하여 알고리즘이 얼마나 오래 걸리는가

공간복잡도

특정한 크기의 입력에 대하여 알고리즘이 얼마나 많은 메모리를 차지하는가

효율적인 알고리즘의 경우,

- 이 둘의 Trade-off 관계가 성립함

```
array = [3, 5, 1, 2, 4] # 5개의 데이터(N = 5)
summary = 0             # 합계를 저장할 변수
```

```
# 모든 데이터를 하나씩 확인하며 합계를 계산
for x in array:
    summary += x
```

```
# 결과를 출력
print(summary)
```

위의 경우, N번의 연산이 실행됨.

표기법 - 빅오 표기법

빅오 표기법	명칭
$O(1)$	상수 시간(Constant time)
$O(\log N)$	로그 시간(Log time)
$O(N)$	선형 시간
$O(N \log N)$	로그 선형 시간
$O(N^2)$	이차 시간
$O(N^3)$	삼차 시간
$O(2^n)$	지수 시간

어떠한 알고리즘이 가지는 최악의 경우에 대한 시간복잡도

- 하지만, 항상 빅오 표기법이 절대적인 것은 아님
(N이 작은 경우에는 최고차항의 영향이 적을 수 있음)

ex) 연산 횟수가 $3N^3 + 5N^2 + 1,000,000$

위 경우, 빅오 표기법으로는 삼차시간이지만, N이 굉장히 적을 때는 최고차항보다 상수항의 영향이 크다.

편의상 앞으로 교재에서 복잡도 = 시간복잡도

일반적으로 삼차시간을 넘어가는 시간 복잡도의 경우, 코딩테스트에서 활용하기 어려움.

문제 풀면서 고려해야할 사항

- 1 알고리즘 문제를 푼다 == 주어진 시간복잡도 내에 문제를 해결한다.
- 2 문제를 풀고 검토, 토론을 할 때 시간복잡도를 더 낮출 수는 없었는가 고민한다.
- 3 BABO 4th 깃헙 자료에 있던 추가 시간복잡도에 관한 자료 업로드 해드릴테니 참고 부탁드립니다.

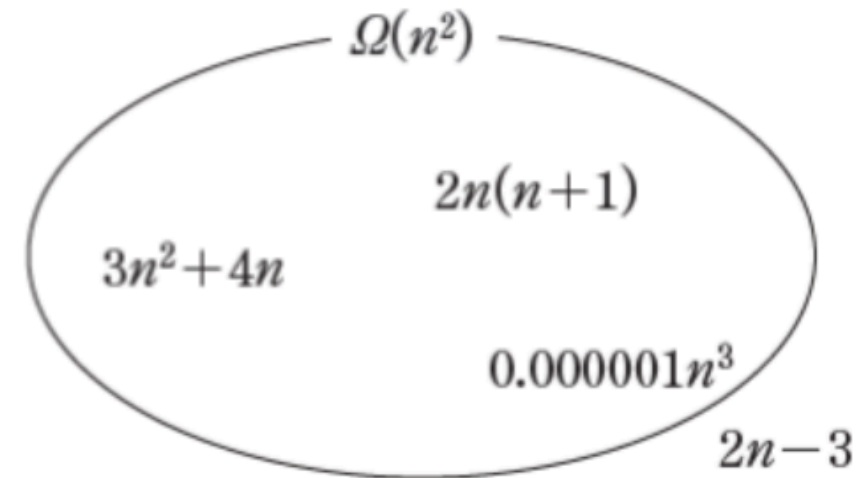
참고

빅오메가, 빅세타 표기법



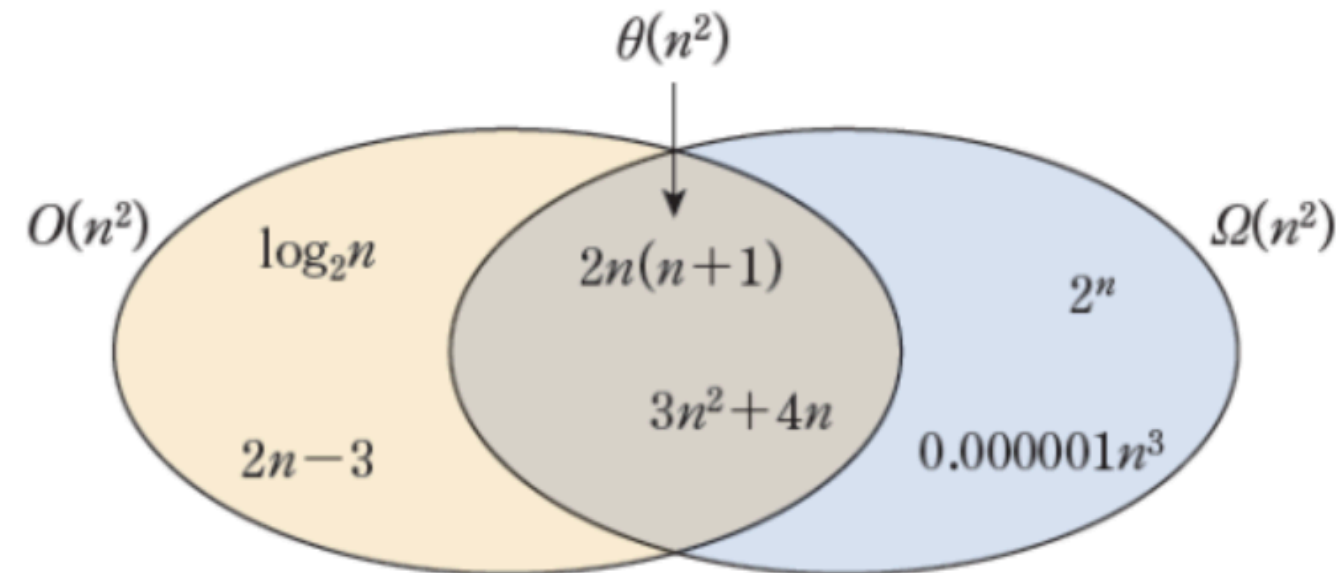
- 빅 오메가: 처리 시간의 하한

$$\begin{aligned}2n^3 + 3n &\in \Omega(n^2) \\ 2n(n+1) &\in \Omega(n^2) \\ 100000n + 8 &\notin \Omega(n^2)\end{aligned}$$



- 빅 세타: 상한이면서 하한

$$\begin{aligned}3n^2 + 4n &\in \theta(n^2) \\ 2n - 3 &\notin \theta(n^2) \\ 0.000001n^3 &\notin \theta(n^2)\end{aligned}$$



출처 : 파이썬으로 쉽게 배우는 자료구조

Q&A

질의 응답



Thank you!