

Lv. 1 가이드

데사 백준 스터디

Lv. 1 구성

- 재귀함수
- 시간 복잡도
- 연결 리스트
- 스택, 큐, 덱
- 정렬 1
- 정렬 2
- 수학 1
- 코딩 1

라이브러리 정렬 알고리즘 사용법

$O(N)$ 정렬 알고리즘

최대공약수, 이항 계수

간단한 브루트포스, 시뮬레이션

얼마나 풀까요?

- **1달에 걸쳐 학습하는 걸 의도하고 만들었습니다.**
 - **빠르게 개념만 익히고 넘어가고 싶다면, 2주 정도를 추천합니다.**
- **이 단계에서 다루는 내용은 앞으로 계속 나옵니다.**
 - **개념을 제대로 이해한 후, 다음 단계로 넘어가는 걸 추천합니다.**
 - **또한 라이브러리 사용법에 익숙하지 않다면, 문제를 많이 풀어보는 걸 추천합니다.**
- **모르는 문제가 있다면 구글에 “백준 + (문제 번호)” 로 검색해보세요!**
 - **친절한 블로그 글을 통해 공부할 수 있습니다.**

재귀함수

- 개념 설명:

- C / C++ : <https://dojang.io/mod/page/view.php?id=584>
- Python: <https://dojang.io/mod/page/view.php?id=2352>

- 이해에 도움이 되는 자료

- <https://blog.naver.com/ingu9981/221651664181>

- 재귀 함수는 자기 자신을 호출하는 함수이지만, ‘자기 자신을 호출’한다는 생각에 집중하면 분석하기 어려워짐
- ‘프로그램이 내가 의도한 대로 동작하는가?’ 에 초점을 맞추는 걸 추천

시간 복잡도

- 개념 설명

- <https://anz1217.tistory.com/7>
- <https://justicehui.github.io/easy-algorithm/2018/03/04/TimeComplexity1/>
- 문제집에 넣은 문제의 지문을 읽고 이해 해 보는 거 추천함

- 시간 복잡도는 Big-O 표기법 등으로 표현할 수 있음

- 전공 수업에서는 정의를 잘 기억해야 하지만,
- 문제풀이에서는 '실행 시간이 이 함수에 비례한다' 정도만 기억해도 무방함

연결 리스트

- 개념 설명

- <https://cheet0se.tistory.com/2>
- <https://hyeinisfree.tistory.com/70>

- 라이브러리:

- C++에서는 `#include <list>`, `#include <forward_list>` 을 통해 사용
- Python에서는 리스트 자료형을 이용해서 문제를 풀 수 있음

- 문제 상황에 따라 직접 구현해서 사용하기도 함

스택

- 개념 설명:

- <https://justicehui.github.io/easy-algorithm/2018/06/03/Stack/>

- 라이브러리:

- C++ : <https://justicehui.github.io/easy-algorithm/2018/06/03/StackSTL/>
- Python: 리스트 자료형을 통해 사용하거나, collections 모듈을 통해 사용

큐

- **개념 설명:**

- <https://justicehui.github.io/easy-algorithm/2018/06/09/Queue/>

- **라이브러리:**

- C++ : <https://justicehui.github.io/easy-algorithm/2018/06/09/QueueSTL/>
- Python: queue 모듈 또는 collections 모듈을 통해 사용

덱

- 개념 설명:

- <https://justicehui.github.io/easy-algorithm/2018/06/12/Deque/>

- 라이브러리:

- C++ : <https://justicehui.github.io/easy-algorithm/2018/06/13/DequeSTL/>
- Python: collections 모듈을 통해 사용

라이브러리 정렬 알고리즘

- 언어별 정렬 라이브러리 사용법

- C++: <https://blog.naver.com/ndb796/221227975229>
- Python: <https://blockdmask.tistory.com/564>

- Bubble Sort, Merge Sort 등 다양한 알고리즘이 있습니다.

- 문제풀이에서는 대부분 라이브러리를 사용합니다.
- 정렬 알고리즘에서 아이디어를 얻어 풀어야 하는 문제도 있기에, 꼭 알아두세요

O(N) 정렬 알고리즘

- Radix Sort

- <https://justicehui.github.io/easy-algorithm/2018/05/02/RadixSort/>

- Bucket Sort

- <https://hroad.tistory.com/25>

- Counting Sort

- <https://justicehui.github.io/easy-algorithm/2018/05/02/CountingSort/>

최대공약수, 이항 계수

- 최대공약수를 구하는 알고리즘으로, 유클리드 호제법이 있습니다.
 - 간단한 설명: <https://justicehui.github.io/easy-algorithm/2018/08/11/GCD/>
 - Proof: <https://rkm0959.tistory.com/179>
- 이항 계수를 구해 봅시다.
 - 파스칼의 삼각형: https://ko.wikipedia.org/wiki/파스칼의_삼각형
 - 파스칼의 삼각형을 $N \times N$ 2차원 배열을 통해 구현해서 이항계수를 구할 수 있음
 - 더 빠르게 구하는 방법이 궁금하다면 다음 글을 참고해주세요:
 - <https://rebro.kr/107>

브루트포스, 시뮬레이션

- **브루트포스 알고리즘**
 - 모든 경우의 수를 검사하는 알고리즘
- **시뮬레이션**
 - 주어진 조건과 절차를 코드로 직접 구현하는 문제

Lv. 1 가이드 vs 백준 단계별로 풀어보기

다루는 내용	백준 단계별로 풀어보기
재귀함수	21단계
시간 복잡도	11단계
연결 리스트	-
스택, 큐, 덱	16단계
정렬 1	13단계
정렬 2	13단계
수학 1	9단계, 19단계
코딩 1	12단계

Lv. 1 가이드 vs 데사 2학년 전공

다루는 내용	데사 2학년 전공
재귀함수	자료구조 1주차
시간 복잡도	자료구조 2주차, 알고리즘 2주차
연결 리스트	자료구조 3주차
스택, 큐, 덱	자료구조 4주차
정렬 1	자료구조 13주차, 알고리즘 1~4주차
정렬 2	자료구조 13주차, 알고리즘 4~5주차
수학 1	-
코딩 1	-