

< 공부 진행 방식 >

[학습 자료]

1. 나동빈 : 이것이 코딩테스트다 + 유튜브 강의
→ 이 책이 알고리즘의 모든 것을 포괄하지는 못하지만, 출제빈도가 가장 높은 알고리즘을 중심으로 구성되어 있음
: 시험 대비 속성서 느낌
2. 파이썬 알고리즘 인터뷰
→ 대학원 연구 수준의 고급 알고리즘까지 다루지는 않지만, 코딩테스트에서 나올 수 있고, 학부생 수준에서 다룰 수 있는 거의 모든 알고리즘이 시각화 자료와 함께 잘 설명되어 있음 + 파이썬 스타일의 코딩 방식에 대해서도 다룸
: 기본에 충실한 개념서 느낌

[공부 방식]

나동빈 책을 이용하여, 공부 페이지를 1회독 / 2+회독 / 기출 문제 풀이 으로 나누어서 공부를 진행하며 이와 동시에, 파이썬 알고리즘 인터뷰를 조금씩 정독해나간다.

< 1회독 >

1회독 목표 : 코딩테스트에 나오는 주요 알고리즘에는 무엇이 있고 어떤 식으로 문제가 나오는지 확인하는 단계

1회독 공부 방법 : 나동빈님 강의 영상을 정주행하면서, 주제마다 아래 적어준 추천 문제 리스트를 풀면서, 알고리즘의 동작 원리에 대해 이해하려는 노력을 하면서 진행한다.

< 2+회독 >

2+회독 목표 : 주요 알고리즘에 무엇이 있는지 파악하였으니, 이제는 문제를 분석할 차례.
코딩테스트 문제는 주로 아래와 같은 세 가지 형태가 있다.

1) 문제 상황을 제시했을 때 (1) 해당 문제 상황을 가장 효과적으로 해결할 수 있는 알고리즘을 찾아내고, (2) 해당 알고리즘을 문제 상황에 맞게 적용하여 문제를 해결하는데, 해결 방식은 보통 “그 알고리즘의 가장 기본 예제” + “추가 조건 처리” + “불필요한 계산 상황 삭제를 통한 효율성 테스트 통과” 로 귀결된다.

(이 유형의 대표 예시 문제 : 카카오 - 양궁 대회 / 카카오 - 표 편집)

2) 메인 알고리즘이 따로 존재하지 않고, 그냥 단순 구현 문제. 거대한 상황을 주고 이를 차근차근 잘 구현해낼 수 있는지를 평가하는 문제

(이 유형의 대표 예시 문제 : 삼성 SW - 톱니바퀴 / 카카오 - 오픈채팅방 / 삼성 SW - 주사위 굴리기 1)

3) 1)의 경우와 2)의 경우가 섞인 문제

(삼성 SW - 주사위 굴리기 2 / 백준 - 뚝딱뚝딱)

이렇게 2+회독은 이제 메인 주제들을 돌면서 각 주제에 대한 문제가 어떻게 구성되는지를 분석하고 약한 부분을 편안하게 코딩할 수 있게 만드는 단계에 해당함

그리고 여기서부터 기업 기출문제 쉬운거 풀어보는 것 나쁘지 않음

〈1회독 문제 추천〉

(준비 여건에 따라서 다 풀거나 일부만 풀거나!, Essential은 반드시 풀 것!)

나중에 스스로 문제 선정할 때 문제 선정 방식 :

백준 -> 주로 알고리즘 유형별로 문제를 풀 때 참고 / 1회독 시에는 해당 알고리즘 문제들 중 쉬운 문제 2-3 + 어려운 문제 2-3문제를 풀어봄

프로그래머스 -> 기출문제 등 종합문제를 수준별로 풀어볼 때 주로 활용

백준 -> 문제집 중 삼성 SW 역량 문제가 퀄리티가 나쁘지 않으니 많이 활용하면 좋음

Essential이라고 적힌 문제 : 해당 알고리즘 및 유형에서 가장 대표가 되는 문제, 이 Essential 문제에서 변형하여 코테 문제를 출제함

1. 구현

(1) 오픈채팅방_카카오 (프로그래머스 Level 2)

<https://school.programmers.co.kr/learn/courses/30/lessons/42888>

(2) 달팽이 (실버 3)

<https://www.acmicpc.net/problem/1913>

(3) 과제는 끝나지 않아! (실버 3)

<https://www.acmicpc.net/problem/17952>

(4) 드래곤 커브 (삼성 SW 역량) (골드 4)

<https://www.acmicpc.net/problem/15685>

(5) 톱니바퀴 (삼성 SW 역량) (골드 5)

<https://www.acmicpc.net/problem/14891>

2. BFS / DFS (가장 빈출)

(유형이 크게 - 영역 구분 (이하 유형 1) / 최단 경로 - 간선 거리 일정 (이하 유형 2) 이렇게 두 유형으로 구성되어 있는데 각 유형의 기본 문제 -> 추가 조건이 추가되는 형태로 문항 구성)

(1) 유기농 배추 (유형 1 Essential) (실버 2)

<https://www.acmicpc.net/problem/1012>

(2) 양 (유형 1 응용, 조건 추가) (실버 1)

<https://www.acmicpc.net/problem/3184>

(3) 빙산 (유형 1 응용, 반복 판단) (골드 4)

<https://www.acmicpc.net/problem/2573>

(4) 인구 이동 (유형 1 응용, 반복 판단) (골드 5)

<https://www.acmicpc.net/problem/16234>

(5) 토마토 (유형 1 응용, 3차원, 반복 판단) (골드 5)

<https://www.acmicpc.net/problem/7569>

(6) 미로 탐색 (유형 2 Essential) (실버 1)

<https://www.acmicpc.net/problem/2178>

(7) A→B (유형 2, 2차원이 아닌 그래프) (실버 2)

<https://www.acmicpc.net/problem/16953>

(8) 벽 부수고 이동하기 (유형 2, 조건 추가) (골드 4)

<https://www.acmicpc.net/problem/2206>

(9) 맥주 마시면서 이동하기 (유형 2, 조건 추가) (실버 1)

<https://www.acmicpc.net/problem/9205>

(10) (도전) 구슬 탈출 2 (삼성 SW 역량) (유형2 + 구현) (골드 1)

<https://www.acmicpc.net/problem/13460>

3. 이분 탐색

이분 탐색은 큰 골격이 항상 똑같고, 중간에 판별 조건 부분만 문제에 잘 맞게 갈아끼면 쉽게 문제를 풀 수 있다. 이를 참고해서 문제를 풀자.

(1) 수 찾기 (Essential) (실버 4)

<https://www.acmicpc.net/problem/1920>

(2) 랜선 자르기 (실버 2)

<https://www.acmicpc.net/problem/1654>

(3) 나무 자르기 (실버 2)

<https://www.acmicpc.net/problem/2805>

(4) 기타 레슨 (실버 1)

<https://www.acmicpc.net/problem/2343>

(5) 공유기 설치 (골드 4)

<https://www.acmicpc.net/problem/2110>

4. Dynamic Programming

이 알고리즘은 깊게 들어가면 난이도가 매우 높은 만큼, 코테에서는 지나치게 어려운 문제까지 출제는 지양하는 편이나, 최근 삼성에서 고난도 문제가 출제된 바가 있는 만큼 충분한 연습이 필요한 파트!

(1) 설탕 배달 (그리디 - 거스름 돈 문제에서 동전이 배수가 안 나오는 경우 → DP) (실버 4)

<https://www.acmicpc.net/problem/2839>

(2) 피보나치 함수 (이 문제를 통해 재귀 방식과 DP 방식의 차이, 각 방식의 장단점을 정확하게 이해 하여야 함) (실버 3)

<https://www.acmicpc.net/problem/1003>

(3) 1로 만들기 (조금 더 복잡한 참조 구조의 DP) (실버 3)

<https://www.acmicpc.net/problem/1463>

(4) 계단 오르기 (1차원 DP Essential) (실버 3)

<https://www.acmicpc.net/problem/2579>

(5) 포도주 시식 (1차원 DP) (실버 1)

<https://www.acmicpc.net/problem/2156>

(6) RGB거리 (2차원 DP Essential) (실버 1)

<https://www.acmicpc.net/problem/1149>

(7) RGB거리 2 (2차원 DP, 조건 심화) (골드 4)

<https://www.acmicpc.net/problem/17404>

(8) 내리막길 (이 문제는 풀다가 너무 힘들면 DFS + DP를 검색해볼 것) (DFS + DP) (골드 3)

<https://www.acmicpc.net/problem/1520>

그 이후 그래프 관련 알고리즘은

백준 -> 알고리즘 분류별 문제로 들어가 해당 알고리즘을 찾고

쉬운 문제 1, 살짝 응용 문제 1

이렇게 2문제 정도 풀어본다!