



Algorithm

〈알고리즘을 대하는 자세〉

2021.03.10 (수)

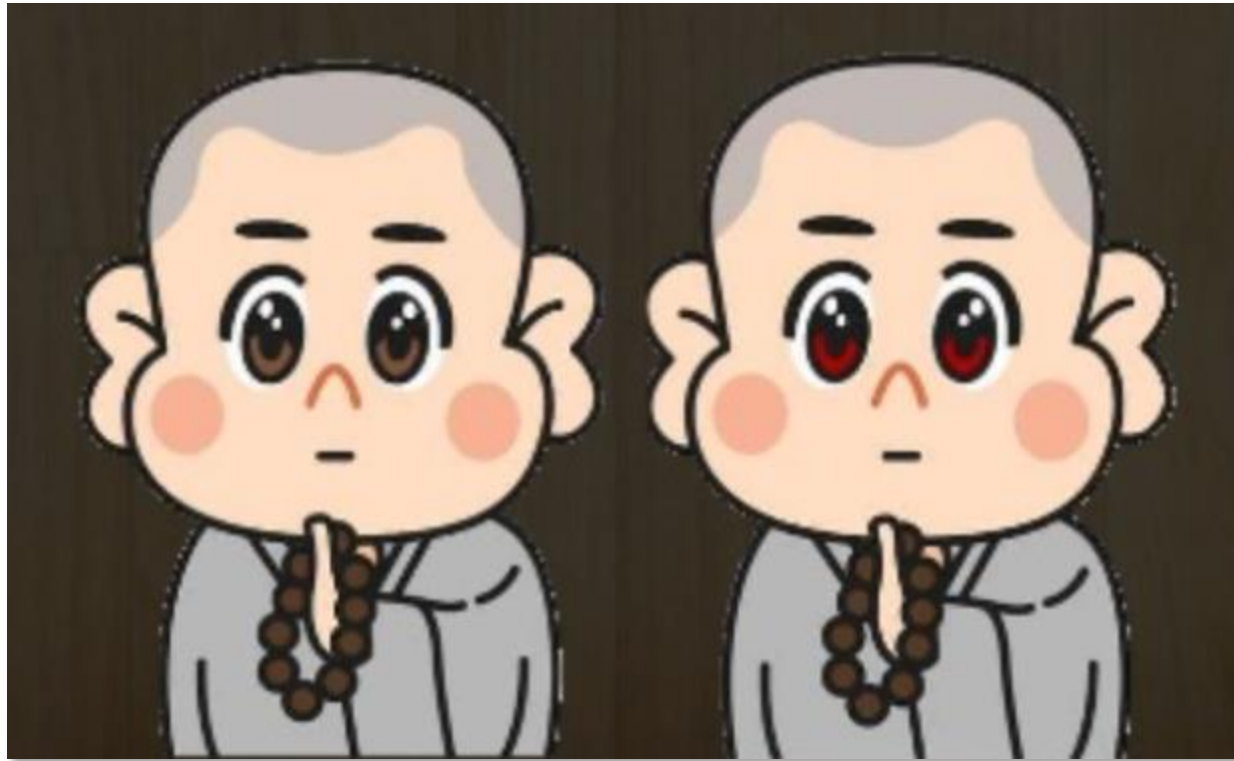
By 박현준 튜터

세션 키워드



모범 공부 방법(싸이클)

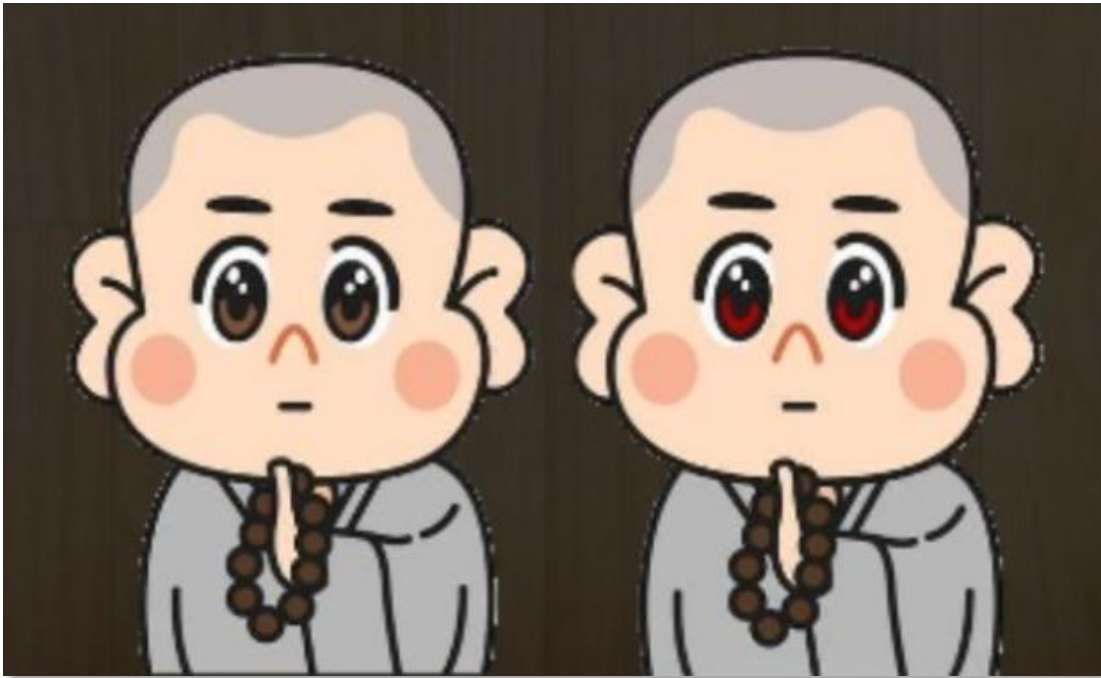
1. 진짜 못하는 사람에게 주는 조언



모범 공부 방법(사이클)

1. 진짜 못하는 사람에게 주는 조언

〈쉬어 가는 코너〉



문제)

1. 승려만 모여 사는 섬이 존재.
2. 빨간 눈, 갈색 눈을 가진 승려들.
3. 자신의 눈이 빨간 사실을 알면 그날 밤 12시에 자살 (강제 마법)
4. 하지만 승려 간에 서로의 눈 색깔 언급 금지의 불문을 존재 상황.
5. 거울 또는 비슷한 물건조차 없기에 자신의 눈 색깔을 스스로 확인은 불가
6. 어느 날, 섬에 관광객이 "당신들 중, '적어도' 한 명은 눈이 빨간색이군요?"
7. 그날 밤부터 그 섬에는 무서운 일이 일어나기 시작

과연 어떤 일이 일어났겠는가?

모범 공부 방법(싸이클)

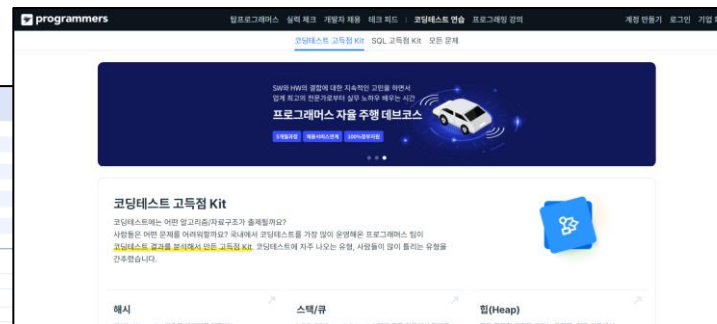
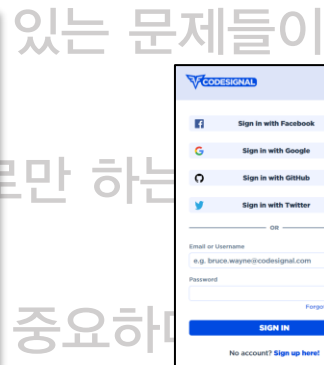
1. 진짜 못하는 사람에게 주는 조언

- ✓ 재미있어야 계속 풀고, 계속 풀어야 성취감이 쌓인다
- ✓ 결국 알아야 풀 수 있는 문제들이 많다. 모르면 당연히 못 푼다
- ✓ 알고리즘은 머리로만 하는 것이 아니다
- ✓ 많이 해보는 것이 중요하다
- ✓ 혼자서 해결하는 능력, 그러나 갈 길은 멀다

모범 공부 방법(사이클)

1. 진짜 못하는 사람에게 주는 조언

✓ 재미있어야 계속 풀고, 계속 풀어야 성취감이 쌓인다



✓ 썰이나 영상 접하기

코딩테스트 플랫폼 이용

모범 공부 방법(싸이클)

1. 진짜 못하는 사람에게 주는 조언

✓ 재미있어야 계속 풀고, 계속 풀어야 성취감이 쌓인다

✓ 결국 알아야 풀 수 있는 문제들이 많다. 모르면 당연히 못 푼다

✓ 알고리즘은 머리로만 하는 것이 아니다

✓ 많이 해보는 것이 중요하다

✓ 혼자서 해결하는 능력, 그러나 갈 길은 멀다

모범 공부 방법(사이클)

1. 진짜 못하는 사람에게 주는 조언

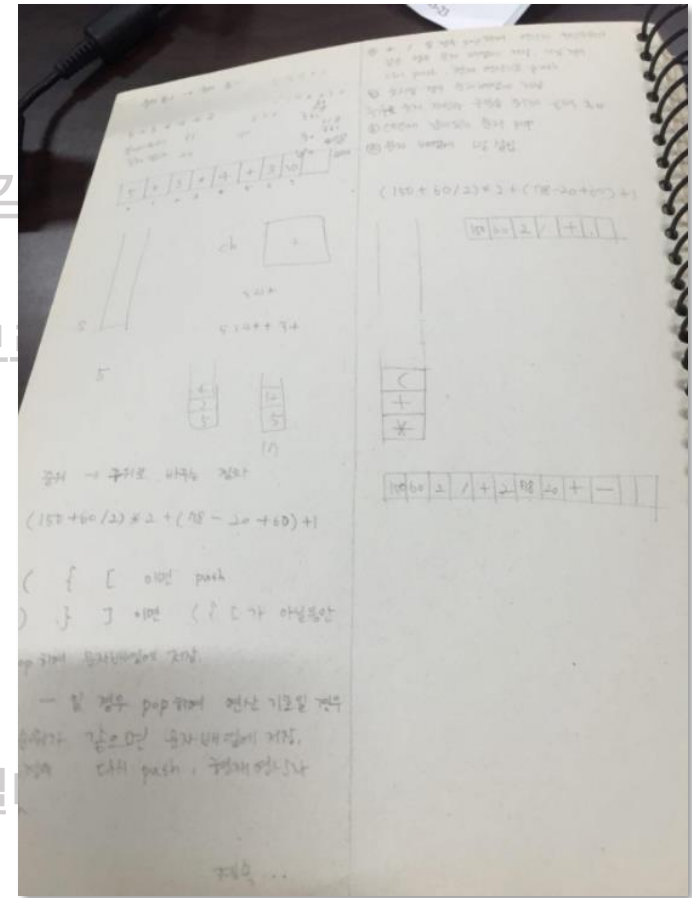
✓ 재미있어야 계속 풀고, 계속 풀어야 성취감

✓ 결국 알아야 풀 수 있는 문제들이 많다. 모

✓ 알고리즘은 머리로만 하는 것이 아니다

✓ 많이 해보는 것이 중요하다

✓ 혼자서 해결하는 능력, 그러나 갈 길은 멀



모범 공부 방법(싸이클)

1. 진짜 못하는 사람에게 주는 조언

- ✓ 재미있어야 계속 풀고, 계속 풀어야 성취감이 쌓인다
- ✓ 결국 알아야 풀 수 있는 문제들이 많다. 모르면 당연히 못 푼다
- ✓ 알고리즘은 머리로만 하는 것이 아니다
- ✓ 많이 해보는 것이 중요하다
- ✓ 혼자서 해결하는 능력, 그러나 갈 길은 멀다

모범 공부 방법(사이클)

1. 진짜 못하는 사람에게 주는 조언

- ✓ 재미있어야 계속 풀고, 계속 풀어야 실
- ✓ 결국 알아야 풀 수 있는 문제들이 많다
- ✓ 알고리즘은 머리로만 하는 것이 아니
- ✓ 많이 해보는 것이 중요하다

✓ 혼자서 해결하는 능력, 그러나 갈 길은 멀다



알고리즘 테스트 공략법

1. 알고리즘을 잘하기 위해서 필요한 3가지 요소

1. 배경지식

2. 구현력

3. 문제해결능력

알고리즘 테스트 공략법

1. 알고리즘을 잘하기 위해서 필요한 3가지 요소

1. 배경지식

2. 구현력

3. 문제해결능력

- 기초적인 프로그래밍 지식과 수학적 지식을 아는 능력
- 부족하면 [어떻게 접근할지 몰라서 솔루션을 열었는데 생판 모르던 외계어가 적혀져 있다] 인 상황
- 제일 공부하기 쉬운 능력이지만 지금 단계에서 가장 중요한 능력

알고리즘 테스트 공략법

1. 알고리즘을 잘하기 위해서 필요한 3가지 요소

1. 배경지식

2. 구현력

3. 문제해결능력

- 본인이 생각한 알고리즘을 그대로 소스코드로 구현하는 과정
- 부족하면 [대충 어떻게 하라는지는 알겠는데 코딩하려니까 한 줄도 못 짜겠다 / 내가 지금 뭘 짜고 있는지 모르겠다 / 코드가 10줄이 넘어가면 엄두가 안난다 / 디버깅을 못 하겠다]를 느낌
- 구현력을 향상 시키기 위해서는 [내가 어떤 프로그램을 만들고자 하는지]를 명확히 해야 한다

알고리즘 테스트 공략법

1. 알고리즘을 잘하기 위해서 필요한 3가지 요소

1. 배경지식

2. 구현력

3. 문제해결능력

- 내가 알고 있는 알고리즘, 자료구조, 다양한 테크닉 등을 지금 당면한 문제에 맞게 변형하여 적용하는 것
- 부족하면 [어떻게 접근해야 할 지 모르지만 솔루션을 열었을 때 아는 알고리즘, 자료구조] 인 상황
- 중위권에서 상위권으로 가고자 할 때 정말 벽을 뚫는다는 느낌으로 노력이 필요한 시점

알고리즘 테스트 공략법

2. 자료구조, 알고리즘 공부의 적정선

- ✓ 코딩테스트에서 보고자 하는 것은 결국 [기초적인 배경지식 + 구현력]
- ✓ [코딩 문법] [시,공간 복잡도 분석]을 아는 것, [배열, 트리, 그래프, 힙, BST, 스택, 큐]와 같은 기본적인 자료구조를 아는 것, [DFS, BFS, 정렬, 백트래킹, DP, 분할정복, 최단거리]까지
- ✓ 응시했던 코딩테스트의 반 이상에서 [DFS, BFS, 분할정복, 최단거리]이 안 나왔음
- ✓ 코딩 테스트는 경시 대회가 아니기 때문에 엄청 어려운 문제는 안나옴. 정 수준의 문제가 나오고 문제가 급각하게 주어졌을 때 (힌트를 주면서라도) 100% 풀수 있는지를 보는 것

모범 공부 방법(사이클)

3. 알고리즘 경험담, 정보, 꿀팁

- ✓ 배X, PXBG, 지X재그 등의 코딩테스트를 통과했는데 80% 이상의 문제가 중~중하 난이도
- ✓ 내준 문제 전부 다 푼다면, 이 회사들에서 코딩테스트 떨어질 일 없다고 단언할 수 있음
- ✓ (면접 꿀팁) 안 중요한 회사 → 중요한 회사 순으로 면접을 보면 면접 연습하다가 자신감이 점점 올라가서 폼이 극에 달한다. 그 때 가장 붙고 싶은 회사 면접을 보면 모든 것을 토해낼 수 있음

모범 공부 방법(싸이클)

4. 알고리즘 테스트 vs 개발 공부

- ✓ 알고리즘을 잘해야 개발을 잘한다? 절대 아니다.
- ✓ 다만 기본적인 구현력과 배경지식이 있어야 회사에 입사할 자격을 얻는다.
- ✓ 그 이후에는 자신의 학습 능력에 따라 개발자의 성장이 달려있다.
즉 시험 패스가 목표이고, 끝이 있는 공부이다. (취업 이직 전에 특히)
- ✓ 알고리즘은 공부법이 존재하고, 전략/공략법을 알고 있으면 도움이 많이 됨
- ✓ (But) 개발 공부는 정말 끝이 없습니다.

알고리즘 Q&A

