

# 알고리즘 공부, 어떻게 해야하나요?

강갓제네럴충무공 박트리 2018. 10. 16. 15:57

---

오랜만에 정상적인 포스팅을 쓴다.

메일로 가장 많이 물어 보는 질문들이 [\[알고리즘 공부 어떻게 해야하나요? 어떻게 하셨어요? 뭘 공부해야 할 지 모르겠어요.\]](#) 와 같은 질문들이다.

위 질문에 가장 심플한 답변은 [\[이런 이런 주제의 알고리즘을 공부하시고 문제를 많이 푸세요.\]](#) 가 끝이다.

사실 이러한 질문과 답변이 모두 우문우답이기 때문에 조금 더 진지하게 초보자의 관점에서 알고리즘 공부를 어떻게 시작하고 어떻게 노력하면 좋을지에 대해 자세히 써볼 예정이다.

---

\*\* 이 글은 [2018-10-31] 에 마지막으로 수정되었습니다. 독자가 읽는 시점의 상황과는 많은 부분이 상이 할 수 있습니다.

\*\* 이 글은 매우 주관적이고 편협하고 얕은 지식으로 작성되었습니다. 본문과 다른 의견이나 틀린 부분이 있다면 메일 또는 댓글로 편하게 말씀해 주세요.

\*\* 이 글은 제 머리속에 있는 생각과 경험들을 두서없이 작성하였습니다. 맞춤법이나 문단 모양과 그림 추가와 같은 부가적인 부분은 능력과 의지가 되는 만큼 하겠습니다.

\*\* 이 글에서 언급하는 알고리즘이란 PS(Problem Solving) 입니다.

\*\* 적어도 하나 이상의 프로그래밍 언어를 알아야 합니다.

---

## \* PS가 뭐예요? 왜 PS 공부해야하나요?

본문에서 언급하는 PS는 현실 세계의 다양한 문제들을 간략화 추상화하여 정의하고, **일반적으로** 어떠한 입력 데이터에 따른 출력 데이터를 특정한 알고리즘으로 구현된 컴퓨터 프로그램을 통해 얻어내는 과정이라고 생각한다. 현실 세계의 다양한 문제에 비해 구하고자 하는 바가 명확하고, 입력 되는 데이터의 특징이 구체적이며 출력하고자 하는 데이터가 명확한 편이다.

최근 기업들에서 PS를 입사 프로세스내에 추가하여 코딩테스트 또는 역량테스트라는 이름으로 시행되고 꽤 높은 중요도를 차지하고 있다. 초중고 소프트웨어 교육에 있어서도 최근 초점이 단순 코딩 방식을 지양하고, 컴퓨팅사고력, 협력적 문제 해결방식을 지향하고자 하는 등 PS에 대한 관심이 높아지고 있다. 그에 따라 많은 수의 알고리즘 대회가 생겨나고 알고리즘 강의 사이트 들이 우후죽순 생겨나고 있다.

그 사람의 구조적 생각능력 문제해결능력 및 구현능력등 다양한 소프트웨어 능력을 단시간에 쉽게 평가할 수 있는 방법이 PS 능력을 보는 것이고 따라서 최근 다양한 기업들이 이 방법을 채택하고 있다고 생각한다. 앞으로 더욱 관심받고 중요해질 PS 분야에 대해서 미리 앞서 나가는 것이 중요하다고 생각한다.

---

## \* PS를 잘하려면 무엇을 공부해야 하는 거예요?

내가 생각하는 PS를 잘하기 위한 3요소를 소개하겠다 아마도 대부분의 기업에서 입사 프로세스로 코딩테스트를 보면서 지원자를 평가하고자하는 이유도 이 3요소가 포함된다고 생각된다.

### 1. 구현력

### 2. 문제해결능력

### 3. 배경지식

\*\* 앞으로 본문에서 계속 언급될 3요소이다.

### 1. 구현력

구현력은 본인이 생각한 알고리즘을 그대로 소스코드로 구현하는 과정이다. 프로그램 순서도를 명확하게 만들고 사용해야 될 변수 나 함수의 데이터 타입등을 올바르게 정하는 과정이라고 생각할 수 있다.

이 능력이 부족하면 문제를 풀 때 [대충 어떻게 하라는지는 알겠는데 코딩하려니까 한 줄도 못 짜겠다.] 라든지 [내가 지금 뭘 짜고 있는지 모르겠다.] [코드가 1000바이트가 넘어가면 엄두가 안난다.] [디버깅을 못 하겠다.] 라는 생각을 느끼게 된다.

구현력을 향상 시키기 위해서는 [내가 어떤 프로그램을 만들고자 하는지]를 명확히 해야한다. [무엇을 입력받아 어디에 저장하고 어떤 과정을 거쳐서 중간 결과로 무엇을 얻고 최종적으로 이런 결과물을 이렇게 출력한다.] 와 같이 순서도를 먼저 적고 이제 디테일 하게 어떤 데이터 타입 또는 자료구조에 저장할 지 생각하는 것을 차근 차근 종이에 적어가면서 연습하는게 좋다.

천천히 적으면서 순서도를 명확히 그리는 연습을 자주 하면 곧 머리속으로 하게 되고 나아가 코딩하면서 동시에 생각하는 수준까지 될 수 있다.

## 2. 문제해결능력

내가 알고있는 알고리즘, 자료구조, 다양한 테크닉등을 지금 당면한 문제에 맞게 변형하여 적용하는 것, 문제를 창의적인 시각에서 접근하여 해결하는 능력은 가장 향상시키기 어려운 능력중 하나이다.

이 능력이 부족하면 문제를 풀 때 [ 어떻게 접근해야 할 지 모른다. ] [그래서 솔루션을 열었는데 내가 아는 알고리즘, 자료구조 ] 인 상황이다.

이 능력을 키우기 위해서는 [양질의 문제(30분 ~ 2시간 고민하면 해결 가능한 문제)를 푸는 것] [이전에 본인이 접근한 다양한 방법들을 잘 정리해 두는 것]이 좋다.

중위권에서 상위권으로 가고자 할 때 발목을 잡는게 결국 문제해결능력이므로 정말 벽을 뚫는다는 느낌으로 노력이 필요한 시점이다.

## 3. 배경지식

기초적인 프로그래밍 문법 및 알고리즘, 자료구조 등을 아는 것 선형대수학과 확률 등과 같이 기본적인 수학적 지식을 아는 것과 같은 능력이다.

이 능력이 부족하면 문제를 풀 때 [ 어떻게 접근할지 몰라서 솔루션을 열었는데 생판 모르던 외계어가 적혀져 있다. ] 인 상황이다.

제일 공부하기 쉬운 능력이고 시중에 있는 거의 모든 알고리즘 역량테스트 대비 강의 또는 책은 배경지식에 대해서 설명하고 있다. 제일 알려주기도 쉬운 능력이다. 다양한 매체를 통해 쉽게 공부할 수 있다.

상위권에 도달하기 위해서는 굉장히 지엽적인 자료구조나 알고리즘을 공부하기도 한다.

대부분의 알고리즘 입문자들은 단지 배경지식만 공부하고 [ 열심히 공부했는데 문제를 왜 못 푸는지 모르겠다. ] [ 나는 뽕대加里인 것 같다 ] 등과 같은 한탄을 한다. [ 강의도 돈주고 사서 듣고 책도 사서 읽었는데 실력이 늘지 않는다. ] 와 같은 고민을 하고 있는 사람이라면 문제는 배경지식이 아니라 다른 능력이 아니었을까? 하고 생각을 해보도록 하자.

이제 대충 본인이 어떤 부분이 부족하고 어떤 부분을 중점적으로 노력해야겠는지 이해가 되는가?

그렇다면 본인이 PS를 왜 공부하는지 그 목표에 따라 공부 계획을 세워보자.

---

## \* 저는 입사 코딩테스트 통과가 목표예요!

내가 치뤘던 입사 코딩테스트는 삼성전자, 카카오, 구글, 라인이고 또 직접 치뤘는지 않은 다른 기업들에 대해서도 여기저기서 들은 바로 생각해 보건데 입사 코딩테스트에서 평가하고자 하는 능력은 [ 기초적인 배경지식 + 구현력 ] 이라고 생각한다.

기초적인 배경지식으로는 지어영말 기본적인 [코딩 문법] [시,공간 복잡도 분석] 을 아는 것, [배열, 트리, 그래프, 힙, BST, 스택, 큐]와 같은 지어영말 기본적인 자료구조를 아는 것, [DFS, BFS, 정렬, 백트래킹, DP, 분할정복, 최단거리] 와 같은 학부수준에서 안 배웠다면 교과과정의 문제가 있을것 같은 기초적인 알고리즘을 아는 것에 해당한다. 추가로 어떤 배경지식을 공부해야 하는지 궁금하다면 [여기](#) 를 참고해 알고리즘 초급, 중급에 해당 되는 내용을 공부해 보자.

온라인 강의나 블로그 또는 서적을 통해 배경지식에 대한 공부를 충분히 했다면 이제 실전으로 문제를 풀어 보자.

삼성전자의 경우 [ [여기](#) 또는 [여기](#) ] 를 풀어 보는 것으로도 충분하다. 시간을 측정하면서 풀어보도록 하자. [몇 분에 문제를 이해했고, 몇 분 동안 풀이를 생각했고, 몇 분동안 코딩했으며, 몇 분동안 디버깅 했는지] 이러한 부분 부분들을 기록하면서 내가 어떤 부분이 구체적으로 부족한지를 인지하고 그 부분에 대해 좀 더 중점적으로 노력하는 것이 필요하다.

카카오의 경우 [ [여기](#) ] 에서 풀어 보는 것이 좋다. 비교적 더 높은 배경지식을 요구하는 경우도 있고, 문자열 파싱 문제가 꽤 많이 나온다.

마찬가지로 위와 같은 방법으로 공부해 보자.

타 기업같은 경우도 매우 고차원적인 배경지식을 요구하고 심오한 문제해결능력을 요구하는 문제를 보지 못했다. BOJ 같은 경우에는 좋은 문제들이 많지만 문제가 난이도 순으로 정렬이 되어 있지 않기 때문에 찾아서 풀기 어려운 구조이다. SWEA 에서 난이도 3 ~ 5 문제를 많이 풀어 보는 방법으로 공부해 보자.

만약 배경지식을 충분히 공부했음에도 불구하고 문제를 접근조차 하지 못하겠다면 본인은 **[컴퓨팅적 사고력]** 이 부족한 상황이다. 문제 해결능력의 하위호환인데 이 능력이 부족하면 **[재귀함수, 완전탐색, 백트래킹, BFS, 분할정복]** 에 대한 이해 자체가 어렵다. 이해하지를 못해서 문제에 적용할 수 도 없는것이다. **[컴퓨터가 1초에 1억번의 연산을 할 수 있음]**을 이해하고 **[따라서 어떤 시간 복잡도 까지는 가능]** 함을 이해하고, **[메모리 제한이 몇 이므로 배열을 어느정도 까지는 할당 할 수 있겠다]** 라는 생각을 할 수 있어야 하고, 재귀함수를 트리형태로 **[어떻게 진입하고 무엇을 하고 무엇을 리턴하고 종료되는지]**를 그릴 수 있어야한다. 완전탐색에 있어서 **[상태공간의 정의]**를 할 수 있어야 하고 **[현재 상태에서 다음 상태로 갈 수 있는 방법이 몇가지 인지]**를 이해하고 **[최종 종료 상태와 최초 진입 상태]** 가 무엇인지 이해하고 그림으로 표현할 수 있어야 한다.

\* 위와 같이 컴퓨팅적 사고력을 기르기 위한 문제 추천은 BOJ 모든 [별찍기]와 [n과 m] 시리즈이다. 또는 SWEA 난이도 1 ~ 2 문제를 많이 풀어 보자.

만약 접근은 충분히 잘했는데 구현하는데 시간이 너무 오래 걸리고 디버깅하는데 시간을 많이 쓴다면, 본인이 자주 한 구현 실수들을 정리하고 대략적인 순서도를 그리고 자주 사용하는 부분을 함수로 템플릿화를 하는 것을 연습하는 것이 좋다. 근본적으로 코드길이가 줄어들어야 실수할 확률이 줄어든다.

\* 디버깅 팁같은 경우는 나중에 여력이 된다면 적어 보겠다.

---

---

\* 저는 삼성전자 SW 역량테스트 등급 취득이 목표예요!

삼성전자 SW 역량테스트는 여기서 설명이 되어있고 A,B,C 형으로 나뉘어져 있다. A형은 입사 코딩테스트와 동일하므로 이 챕터에서는 B 형과 C형에 대해서만 집중적으로 다룰것이다.

이 챕터는 내용이 길어질 것으로 예상되므로, 따로 작성하겠다.

## 삼성전자 SW 역량테스트 B형, 어떻게 공부해야하나요?

삼성전자 SW 역량테스트 C형, 어떻게 공부해야하나요?(아직 작성안함 곧 쓸꺼임)

---

---

\* 저는 PS 대회나가서 수상하는게 목표예요! 그냥 PS 재밌어요!

대학생, 일반인 및 초중고등학생을 포함해서 국내 또는 세계적인 대회에 나가고, 수상하는게 목표인 사람들을 위해 적어보겠다.

\*\* 앞으로 Codeforces 레이팅 기준을 많이 사용할 건데 코드포스는 이런 사이트 이고 주기적으로 대회가 올라와서 전세계 사람들과 경쟁하고 레이팅을 매겨주는 시스템이다. 몇 번 참가해보면 대략적인 난이도와 이 정도 레이팅이면 어느정도 수준이구나를 알 수 있으므로 기준으로 사용하겠다.

나는 2000점 쯤으로 보라돌이 중간따리이다. 이미 오렌지 이상의 실력을 가진 분들은 너그러운 마음으로 읽어주길 바란다.

그리고 내 생각에 보라돌이 ~ 오렌지 와리가리 하는 실력이 국내 대학생 큰 대회(SCPC, KAKAO 등) 에서 수상권이라고 생각한다.

대상 독자는 민트 지박령, 민트 ~ 블루 왔다갔다랑 블루 지박령, 블루 ~ 퍼플 와리가리 하는 사람들이다.

충분히 많은 코드포스 대회를 돌렸는데 **그린** 이하에 박제가 되어버렸다면, 아직 영어가 너무 부족하거나 기본적인 코딩능력 및 사고력이 부족하므로 좀 더 쉬운 내용을 찾아보고 공부해보는 것이 좋다고 생각한다.

---

**대회하려면 C++ 쓰세요, 보조무기로 Python 쓰세요.**

일단 시작은 **종만북** 사세요이다.

배경지식을 늘리는 가장 쉽고 좋은 방법이다 물론 관련 키워드를 블로그를 통해 공부하는 법도 있지만 내가 늙어서 그런가 종이로 보는게 더 학습률이 좋았던 것 같다.

종만북은 처음 읽기 너무 어려운 책이다. 보통 쪽 읽다가 동적계획법2에서 무너지게 된다. 나또한 그랬고

추천하는 챕터 읽는 순서는 [1-8, 10, 14, 16-19, 21-25, 27-32] 먼저 읽고 나머지 부분들은 천천히 읽어보면 된다. 이정도 배경지식만 있어도 퍼플을 가기에 부족함이 없고 추가적으로 알아 둘만한 것은 [Sparse Table, MCMF, Lazy Propagation] 정도가 있을 것 같다.

이제 구현력과 문제해결능력을 동시에 늘리기 위한 가장 쉬운 방법은 **[백준 랭작]**을 하는 것이다. 가장 주먹구구 방법이기 때문에 시간이 넘치는 사람들만 추천한다. 나같은 경우에는 **이곳** 에서 하루에 최소 5문제씩 계속 꾸준히 풀었다. 알고리즘 유형을 알고 문제를 푸는 것과 모르고 푸는 것은 큰 차이가 있고, 매일 풀다보니 시간 짬짬히 남을 때 계속 문제에 대한 고민을 할 수 있어서 학습량이 많았다.

문제를 충분히 많이 풀어서 구현력은 어느정도 자신은 있는데 코드포스만 보면 **민트**로 수렴한다면 이제 양질의 문제를 풀어서 문제해결능력을 좀 더 키워야 하는 시점이다. 문제해결능력은 위에서 언급했던 바와같이 적당한 시간을 생각하고 푸는 문제를 많이 풀어야 된다.

**[유사코 실버, COCI, 앳코더 레귤러, 코포Div2-3]** 등의 문제를 푸는 양을 늘리고 백준 랭작량을 좀 줄이면 되겠다.

COCI, 앳코더, 코드포스 같은 경우에는 문제가 난이도 순으로 있기 때문에 보통 연습할때 대회 시간내에 풀 수 있는 만큼 풀고 항상 딱 한 문제씩만 더 풀기라는 목표를 가지고 꾸준히 노력해보자.

이렇게 공부를 하다가 **블딱이**가 되고 정체기가 되었다면, 본인의 코포 현황을 분석해보자. 시간을 풀로 다쳐서 3-4 문제를 풀고 블루 정체인지 엄청 빠르게 2-3문제를 풀고 정체인지 풀었는데 시스템에서 다터지는지, 1번 상황은



구현력을 더 늘려야 되는 시점이다. 다시 랭작 시작하자.

2번 상황은 못 푼 문제의 솔루션을 봤을 때 그 알고리즘이 아는건지 모르는건지 체크하고 접근이 중요한 문제였다면 양질의 문제를 푸는것을 많이 해보자. COCI, 앳코더, 코포 버추얼을 돌기보다. [ Div2 C ~ D에 해당하는 문제를 꼭 뽑아서 계속 시간재면서 풀어 보는 것]이다. 나 같은 경우에는 3번 상황이었는데 코딩할 때 생각을 좀 하면서 코딩하면 비교적 제일 쉽게 해결된다. 예외가 먼지 가장 작을 때 가능한 지 가장 클 때 가능한지 지금 이게 가능한 그리디인지 와 같은 생각~

코드포스 Div2에서 D까지 빨리 풀거나 시간 채워서 E까지 풀게 된다면 어느순간 **보라돌이**가 되어있을 것이다. 이제 보라돌이 부터는 Div1 문제를 풀 수 있게 된다. 따라서 좀 더 고급 알고리즘과 자료구조에 대한 배경지식에 대한 공부가 필요한 시점이 온다. 이 부분에 해당하는 양질의 책은 잘 모르겠고 보통 구글링을 통해서 공부하였다. 다양한 고급 알고리즘 키워드 들을 구글링하여 일주일에 1개 정도 공부하고 연습문제 풀어보는 식으로 익혀보면 좋을 것 같다.

문제해결능력을 키우기 위해서는 [ 유사코 골드, IOI, 앳코더 그랜드, 코포Div1, JOI, KOI, 각종 지역 리저널 및 WF ] 등과 같은 문제를 시간을 많이 가지고 풀어 보는 것이 중요하다.

보통 성장 과정이 [시간이 무한대가 있어도 CDE못 품 -> 시간이 있으면 푸는데 대회중에 못 품 -> 대회중에 풀 때도 있음 -> 대회중에 품] 순이다.

Div1에서 보통 매우 빠르게 AB를 빠르게 풀거나 시간을 다써서 AB+[C,D,E] 를 성공한다면 **오렌지** - **레드**에 진입할 수 있다.

블루 이상이라면 충분히 국내 프로그래밍 대회 본선을 뚫을 정도는 되기 때문에 다양한 대회에 나가면서 경험을 쌓고 멘탈을 잘 지키도록 하자. 본선에서 다양한 사람들을 보면서 친해지는게 본인의 학습의욕에도 더 도움이 되고 임의의 라이벌을 정하는 것도 도움이 될거라고 생각한다. 이후 꾸준히 공부하면서 퍼플-오렌지 구간에 도달해 수상권이 되어 보도록 하자.

---

나는 처음에 동아리도 없고 정보도 없어서 **백준 Slack**을 통해서 정말 많은 도움을 받았고, 혹시 본인 또한 그러한 상황이라면 Slack을 통해 많은 것을 물어보도록 하자 물론 [ 상대방은 아무런 대가 없이 본인에게 시간과 노력을 할당하는 행위 ] 임을 인식하고 서로 존중하도록 하자 **카카오톡** **오픈 채팅** 또한 유용하다.

2020. 1. 6.

알고리즘 공부, 어떻게 해야하나요?

이 포스팅은 아무런 정보없이 알고리즘 공부의 압박을 느끼다 비싸기만한 강의를 수강하고 좋은 결과를 얻지 못하는 많은 사람들을 위해 선의로 작성되었으므로 댓글 하나라도 적고 가도록 하자