

# 0x00 강의소개



- 이 강의는 삼성전자 SW Test A형과 B형의 중간 수준의 코딩테스트에 나올 수 있는 자료구조 및 알고리즘을 알려주고, 실제로 코딩테스트를 잘 넘길 수 있도록 도움을 주기 위한 강의입니다.
- 이미 코딩테스트를 걱정하지 않으셔도 될 실력을 가지신 분은 굳이 강의를 들으실 필요 없이 그룹에 들어오셔서 문제만 풀어보셔도 됩니다.(그룹 링크는 뒤에 소개가 됩니다.)
- 코딩테스트에서 나오지 않을 부분은 다루지 않을 예정이기 때문에 추후에 면접을 대비하기 위해서는 별도로 학부 과정의 알고리즘 강의 커리큘럼을 따라서 공부를 하시는 것을 추천 드립니다. (Binary Search Tree, AVL Tree, Linked list, Hash collision, etc ...)

# 0x01 요구되는 선수 지식



- 강의는 C++로 진행이 되기 때문에 C/C++ 문법을 모르면 강의를 따라올 수가 없습니다. (단, 다른 부분은 다 괜찮은데 C++의 클래스와 STL만을 모른다면 클래스와 STL을 완벽하게 익히고 이 강의를 듣는 것 보다는 강의와 함께 C++ 공부를 병행하는 것을 추천 드립니다.)
- 지금까지 C++을 모르고 JAVA, Python으로 코딩테스트를 준비했다면 두 언어는 C++에 비해 느리고, 자료의 양이 C++에 비해 부족하기 때문에 이 기회에 C++을 배워보는 것을 추천 드립니다. 특히 Python은 정말 느리기 때문에 **Python만을 알고 계신다면 반드시 C/C++을 공부하셔야 합니다.**

# 0x02 알고리즘이란 무엇인가?



- 알고리즘 = 주어진 문제를 해결하는 절차 및 방법
- 알고리즘 문제를 푸는 공부를 PS(=Problem Solving) 혹은 CP(=Competitive Programming)이라고 부르기도 합니다.
- 코딩테스트는 PS, CP와 같이 주어진 문제를 정해진 **시간 제한**과 **메모리 제한**내로 해결할 수 있는 능력을 측정하는 테스트입니다.

## The Feynman Problem Solving Algorithm:

**1. Write down the problem.**

I  
V

**2. Think real hard.**

I  
V

**3. Write down the solution.**

# 0x02 알고리즘이란 무엇인가?



- 알고리즘을 잘 한다는 것은 아래의 3가지 능력을 모두 갖추고 있다는 의미입니다.



배경지식

다양한 알고리즘, 자료구조, 기타 테크닉 등과 같이 문제를 해결하기 위해 필요한 지식



문제해결능력

배경지식을 지금 당면한 문제에 맞게 잘 변형해서 적용시키는 능력



구현력

본인이 생각한 알고리즘을 코드로 잘 옮겨낼 수 있는 능력

- 반대로 말하면 코딩테스트에서 좋은 성과를 내기 위해 세 가지 능력을 모두 가지고 있어야 한다는 의미입니다.(다만 코딩테스트의 경우, 알고리즘 대회에 비해 전형적인 문제를 많이 내므로 문제해결능력의 중요성이 조금 낮습니다.)

# 0x02 알고리즘이란 무엇인가?



- 이 강의를 통해 배경지식과 강의에서 다루게 될 일부 전형적인 문제에 대한 문제해결능력을 길러드릴 수는 있지만 시간이 허락한다면 직접 다양한 문제를 풀어보며 구현력과 문제해결능력을 기르셔야 합니다.
- 강의에서 풀이를 소개할 문제 이외에도 별도로 풀어보면 도움이 될 다양한 문제를 소개해드릴 계획입니다.



# 0x03 환경 세팅



- C++ 컴파일을 위해 사용할 수 있는 프로그램으로는 Visual Studio 2017, Visual Studio Code, Code::Blocks, Dev-C++ 등이 있고 저는 Windows에서 Visual Studio Code + MinGW를 사용합니다.
- 코딩 테스트의 채점 서버는 99% GCC를 기반으로 하므로 로컬에서도 GCC를 기준으로 환경을 만들면 환경 불일치로 인한 오답을 방지할 수 있습니다. (문제에 대한 다른 사람의 정답코드를 직접 돌려볼 때도 마찬가지로 이유로 GCC에서 돌리는게 낫습니다.)

```
int main(void) {  
    int n = 5;  
    int arr[n] = { 1, 2, 3 };  
}
```

int n  
식에 상수 값이 있어야 합니다.  
변수 "n" (선언된 줄 6) 값을 상수로 사용할 수 없습니다.

Visual Studio 2017

```
int main(void) {  
    int n = 5;  
    int arr[n] = { 1, 2, 3 };  
}
```

Visual Studio Code

## 0x03 환경 세팅



- VS2017이 초심자에게 가장 익숙한 프로그램이겠지만 GCC 컴파일러와 다르게 동작하는 명령들이 있으므로 이번 기회에 Visual Studio Code로 갈아타는 것을 추천 드립니다. (Code::Blocks, Dev-C++도 괜찮지만 별다른 이유가 없으면 사용자가 많은 VS Code를 추천 드립니다.)
- Baekjoon Online Judge 그룹에 가입신청을 해주세요.  
(<https://www.acmicpc.net/group/member/4490>)

# 0x04 알고리즘의 중요성?



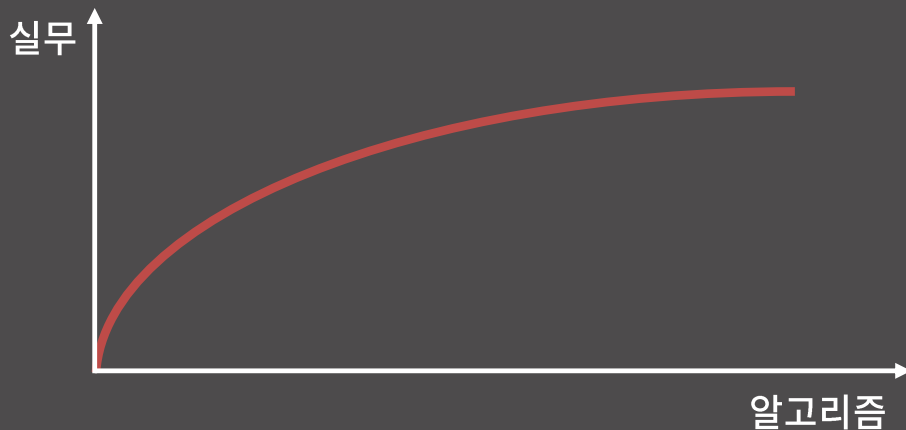
- 알고리즘이 실무에 얼마나 큰 기여를 할지는 저도 잘 모릅니다.(아직 대학생이라 ㅎㅎㅎ)
- 그렇기에 현업에 계신 분들이 알고리즘의 중요성을 평가절하하는 것도 일리가 있다고 생각합니다. (또 분야에 따라 알고리즘이 필요한 정도도 차이가 많이 날 것입니다.)
- 그렇지만 앞서 말한 배경지식, 문제해결능력, 구현력이 높아서 나뉘는 없습니다.
- 그리고 기업에서 알고리즘 능력을 평가하지 않던 시절, 간단한 정렬조차 제대로 활용하지 못해 비효율적인 코드를 만들어내는 경우가 비일비재했습니다.
- 기업들이 코딩테스트를 진행하는 것도 코딩테스트를 통과할 수준의 배경지식, 문제해결능력, 구현력은 갖추고 있기를 원하는 것이 아닐까요?



## 0x04 알고리즘의 중요성?



- 굳이 따지자면 알고리즘 실력과 현업에서의 능력은 log scale을 따르는 것 같습니다.



- 알고리즘을 공부하는게 즐겁다면 계속 공부를 하시면 되고, 그렇지 않다면 코딩테스트에 합격할 정도의 수준만 만들어 놓은 뒤 굳이 알고리즘에 연연할 필요 없이 하고 싶은 공부를 하시면 될 것 같습니다.(해킹, 블록체인, AI 등등...)