유한개의 명령어들의 순차적나열이다.
기계가 이해할수 있을 정도로 문제를 잘게 쪼개어 나열하기만 하면
컴퓨터는 그것을 자동으로 처리가능하다.
알고리즘은 매우 일반적으로 만들어야 한다.

일반적으로 구조로부터 알고리즘이 만들어진다.
알고리즘을 만드는 것은 사람이 한다.

문제에서 문제 해결 방법을 배우는 것, -> 대부분의 학문에서 이렇게 함
알고리즘을 만드는 것은 요즘 어떠한 전공이라도 할줄 알아야 할 것이다.
알고리즘을 컴퓨터 프로그램으로 만드는 것은 컴퓨터학과들이 많이 한다.

10만 종업원의 월급계산?
사장이나 임원이나 인사 전문가가 문제해결 방법을 제시
알고리즘으로 변환은 프로그래머들이 많이 했는데,
사실 요즘은 그 도메인 전문가가 그것을 하는 것이 맞다.
그 알고리즘으로 컴퓨터 프로그램을 만들어 운영하는 것은 다른 사람이다.

만약에 문제가 생긴다면?
-> 컴퓨터가 아닌, 사람의 실수, 의도이다.
컴퓨터에게 모든 것을 돌리지 말자는 것이다.

알고리즘을 만들어 보는 과정은 나의 생각에 대해 다시 스스로 생각해보게 하는 과정이다.
나의 생각을 개선 시켜나갈 수 있다.

알고리즘을 어떻게 표현을 해야하나
step-by-step의 나열인데, 유한한 시간내로 그것이 끝나야 한다.

모두에게 적용가능한 입력(input)이 들어가야 한다.
즉 input이 들어왔을 때, 원하는 output을 출력하는 절차를 순차적으로 표현하는 것이다.

이 표현은 크게 순서도와 의사코드가 있다

으로 표현가능
그림으로도, 말로도 표현가능한 것이다.
하지만 이런 표현은 "모든 상황"을 다 정확하게 표현해야 한다.

instrution의 종류에는 "조건이 맞을 때까지 반복", "조건에 따라서 분기" 등이 있다.
while과 branch는 알고리즘을 구성하는데 매우 효율적이다

외국 요리책은 명확하게 되어있는 경우가 많다.

사실 일잘하는 사람은 다 알고리즘적으로 행동하고 있었다.

문제:
목욕탕에서 들어오는 손님의 성별에 따라 남탕, 여탕으로 들어가는 다른 입구를 안내해준다.
-> 반복적인 일이라서 컴퓨터 알고리즘으로 만들어서 로봇에게 시키려고 함.
알고리즘:
input (사람)-> 1.인사, 2. 얼굴확인, 3. 남여 확인, 4. 안내 ->output(여탕 or 남탕)
반복
↳이것을 보통 기계학습으로 처리했다.

알고리즘 만들기

1) 내가 이미 잘 하고 있는 것을, 그리고 어떻게 하는 것인지도 잘 알고있는 것을
   알고리즘으로 표현
2) 나는 잘 못하지만 다른 사람이 잘하는 것을 관찰해서/파악/분석해서
   알고리즘으로 변환
3) 나는 잘하고 있지만, 내가 어떻게 하고 있는지는 모르는 것에 대해,
   알고리즘을 만들어야 함

고대 바벨론 점토판에서도 알고리즘이 있었다.