어바웃 IT 기술사 11

IT 기술사 과목별 소개
- 알고리즘

글. 강자원 컴퓨터시스템응용기술사
KBS MNC(Media Network Center)팀 (jwings@kbs.co.kr)

연재 목차

우리는 일상 속에서 목적을 달성하기 위해 많은 선택과 결정을 한다. 그 선택과 결정에는 각자 나름의 규칙과 방식이 존재한다. 학습에 의해 가장 적합하다고 판단해서 혹은 단지 기분에 따라 무작위로. 중요한 것은 처한 상황에서 어떤 선택이나 결정을 하느냐에 따라 결과가 달라진다는 사실이다. 사전적 의미로 알고리즘이란 '수학이나 컴퓨터과학 분야에서 컴퓨터가 특정한 문제를 해결하기 위해 기술한 과정과 절차'를 말한다. 여기서, 문제해결은 문제 상황이 원하는 기준이나 조건을 만족하여 해결상황으로 되었을 때를 가리킨다. 좀 더 알고리즘을 학문적으로 말하면, 시작 문제 상태에서 종료 해결 상태까지 전이 과정을 절차적으로 기술한 문장들을 말한다. 알고리즘은 '컴퓨터를 이용한 문제해결'을 위해 필수 불가결한 것이다. 그러나 비단 컴퓨터를 이용한 문제해결에만 국한되지 않는다. 사람이 주체가 되는 업무에서도 그 업무 과정이 명확하게 기술되지 않고서는 원하는 결과를 만들어낼 수가 없다. 현재 상태에서 원하는 목표 상태로의 변화를 꾀하기 위해서는 알고리즘을 알아야만 한다.

인공지능, 알고리즘, 기계가 우리 실생활에 점점 더 깊숙이 들어오고 있다는 걸 최근 몇 년 사이에 피부로 더 많이 느끼고 있다. 음식점 같은 경우에도 홀 직원 없이 발권기로 주문하는 매장이 많아졌고 주차 시스템도 점점 무인화되어 가는 추세이다. 처음엔 낯설었지만, 점점 더 인간의 삶에 친숙히 다가와 있는 알고리즘. 우리는 이 알고리즘을 어떻게 정립시켜 나가야 할까?

알고리즘 그리고 자료구조, 왜 중요할까?

자료구조란?

컴퓨터에서 처리할 자료를
효율적으로 관리하고 구조화 하는 방법

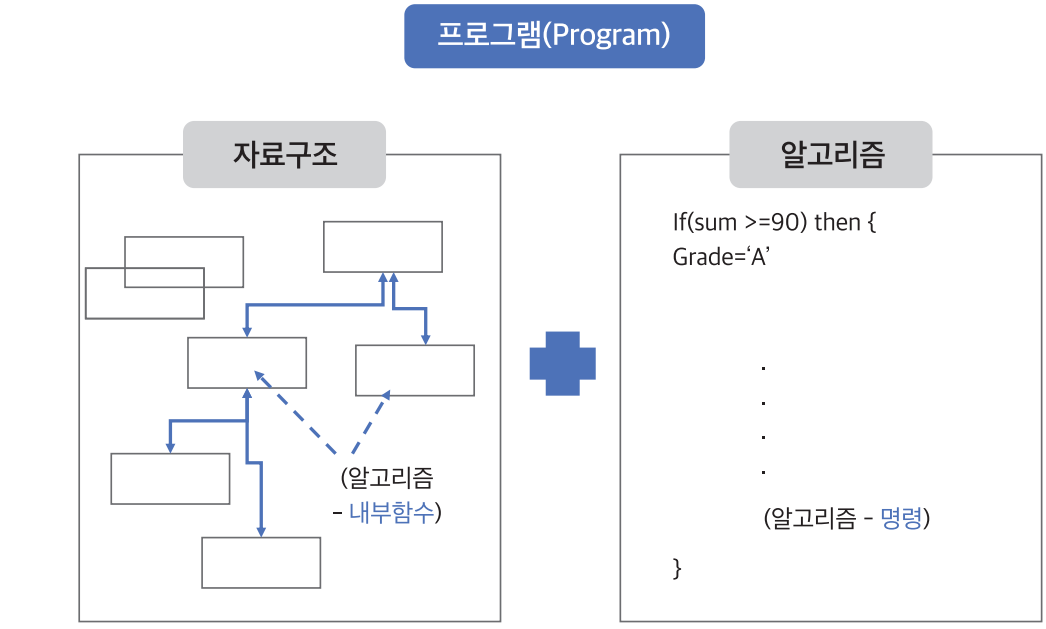
알고리즘이란?

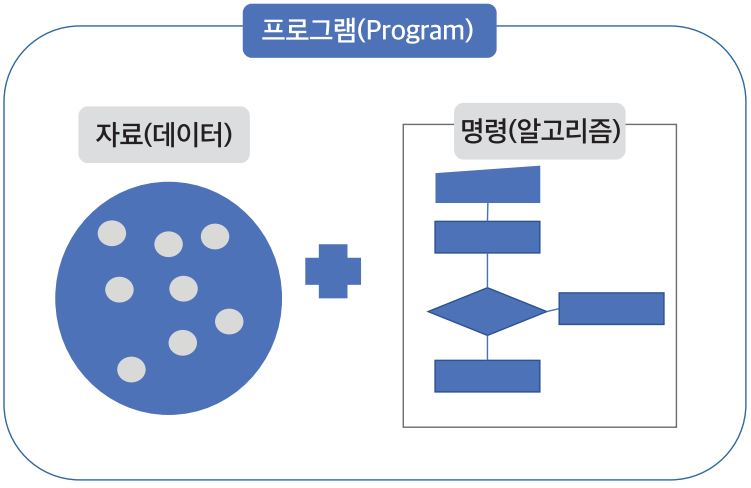
프로그램의 처리 순서나 절차를 작성한 것

알고리즘 특정 목적 달성을 위한 절차

자료구조 알고리즘에 필요한 데이터의 집합

알고리즘을 이야기할 때 빠지지 않고 등장하는 개념이 자료구조다. 대부분의 컴퓨터 프로그래밍은 '자료구조 + 알고리즘'이라 말할 수 있다. 알고리즘을 '특정 목적 달성을 위한 절차'라고 한다면 자료구조는 '알고리즘에 필요한 데이터의 집합'이라 할 수 있다. 다음 그림은 자료구조와 알고리즘의 관계에 대해 설명한 것이다.





프로그래밍이란 코딩을 통해서 어딘가에서 받은 데이터를 가공하여 또 다른 데이터를 만들어 어딘가로 전달하는 과정이라 볼 수 있다.

그러니까 공장에서 제품을 만들 때 재료를 가공하여 다음 공정으로 넘기는 과정과 정말 유사하다. 그렇다면 무엇이 중요할까?
바로 생산 효율성이다. 프로그래밍에서 자료구조와 알고리즘이 이제

왜 중요한지 이해가 될까?
프로그래밍이든 제품 제조든 효율성을 위해서는 시간과 공간의 최대활용이 중요하다. 프로그래밍과 실제 제품 제조에 있어서 효율성을 위해 가장 중요한 것은 시간 복잡성과 공간 복잡성이다. 시간복잡성은 쉽게 말해 자료(재료)가 2차 자료(완성품)로 되는 데까지 소요되는 시간이 얼마나 걸리는지, 얼마나 단축할 수 있는지 등을 의미하는 말이다. 다음으로 공간복잡성은 자료(재료)가 2차 자료(완성품)로 되는 데까지 필요한 장비(프로그래밍에서는 메모리) 사용이 얼마나 필요한지, 또 얼마나 단축할 수 있는지를 의미하는 말이다. 제품 제조에서 제조 공정의 최적화를 위해 시간과 공간에 대한 고민을 하듯이 프로그래밍에서도 가장 효율적인 방법으로 최고의 결과물을 만들어내는 코딩을 구성하는데 필요한 것이 자료구조와 알고리즘인 것이다.
조금 더 와닿기 쉽게 말해볼까?

그림과 같은 화면을 볼 때 기분이 어떤가? 나의 소중한 시간을 빼앗기고 있는 것 같지 않은가? 자료구조 알고리즘을 잘 아는 사람이 프로그래밍을 개발한다면 그림과 같은 화면을 보는 시간을 줄일 수 있다. 이미지를 보고 다음과 같이 분석할 수 있다.

컴퓨터가 처리해야 할 자료가 많고 복잡하거나 (자료구조)
자료의 처리 과정이 길거나 (알고리즘)
컴퓨터의 메모리가 부족하면 (공간복잡성)
정말 오래 지속할 것이다. (시간복잡성)

우리는 이 문장에서 자료구조와 알고리즘 외에 컴퓨터 구조적인 부분까지도 들여다볼 수 있다. 이처럼 좋은 프로그램을 개발한다는 것은 많은 부분을 알아야 하고 인프라적인 환경 상황까지 고려해야 한다는 것이다.

알고리즘, 인간의 사고방식을 이해하다

방을 정리한다고 해보자. 방에 널려 있는 물건의 양에 따라 정리에 걸리는 시간이 달라질 것이다. 프로그램을 작성할 때에도 입력의 크기에 따라서 프로그램이 계산하는 횟수가 크게 달라진다. 입력된 자료의 양과 알고리즘 실행에 걸리는 시간 사이에는 어느 정도의 관계가 있다. 이것을 알고리즘의 시간 복잡도라 한다. 이해하기 쉽게 설명하기 위해 한 가지 프로그램으로 예를 들어보자. 다음 페이지의 두 프로그램은 자연수 N이 주어졌을 때 1부터 N까지의 합을 구하는 프로그램들이다.

소스 1에서는 for 문을 활용하여 직접 res에 1에서 N까지를 더해서 합을 계산했다. 그리고 소스 2에서는 1에서 N까지 자연수들의 합은 n(n+1)/2인 수학 공식을 이용하여 합을 계산했다. 소스 1의 경우에는 N=1,000이라면 덧셈이 1,000번 일어나고, N=1,000,000이라면 덧셈이 1,000,000번 일어나게 된다. 계산하는 횟수가 N에 비례하는 것이다.
반면에, 소스 2에서는 N=1,000이든, N=1,000,000이든 무조건 덧셈 한 번과 곱셈 한 번, 나눗셈 한 번만을 하게 된다. N에 아무 관계 없는 상수 시간 안에 계산이 처리되는 것이다.

모든 알고리즘은 이처럼 입력되는 데이터의 크기 또는 개수에 따라 이의 계산 횟수와 수행 시간이 크게 달라지게 된다. 입력받는 데이터의 크기에 따른 알고리즘의 수행시간의 변화가 시간복잡도이다. 시간복잡도가 더 큰 알고리즘들은 더 큰, 혹은 많은 데이터를 처리할 때 훨씬 더 오랜 시간이 걸리게 된다. 그런데 이 계산 횟수의 정확한 측정이 어려우므로 보통 Big O 표기법이라는 걸 이용해서 표시한다. 시간복잡도를 계산하는 방법까지 여기서 설명하긴 어렵지만

소스#1

C언어

```c
#include <cstdio>

int main(void)
{
        int N, res = 0;
        scanf("%d", &N);

        for (int i=1; i<=N; i++) {
                res += i;
        }
        printf("%d₩n", res);

        return 0;
}
```

Python

```python
res = 0
N = int(input())

for i in range(1, N+1):
    res += i

print(res)
```

소스#2

C언어

```c
#include <cstdio>

int main(void)
{
        int N, res = 0;
        scanf("%d", &N);

        res = N * (N+1) / 2;
        printf("%d₩n", res);

        return 0;
}
```

Python

```python
N = int(input())

res = N * (N+1)//2

print(res)
```

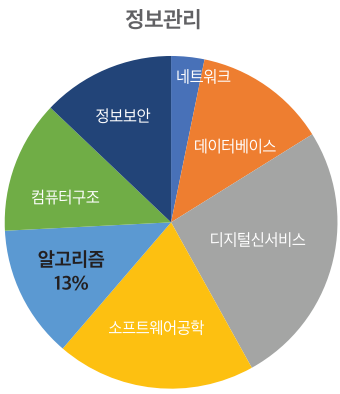
어쨌든 알고리즘은 이런 형식으로 평가하게 된다.

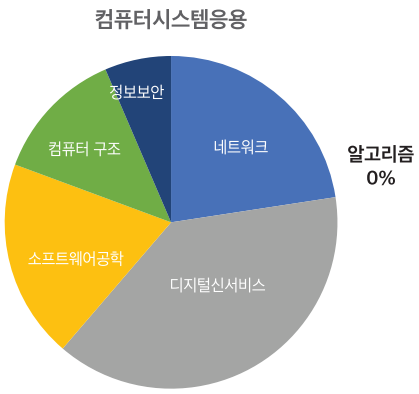
위의 소스들의 시간 복잡도는 어떻게 쓸 수 있을까? 소스 1은 O(N)의 시간 복잡도를, 소스 2는 O(1)의 시간 복잡도를 가지게 된다는 것을 알 수 있다. 직관적으로도 알 수 있듯이, O(1)의 알고리즘이 O(N)의 알고리즘보다 더 빠른 알고리즘이라고 할 수 있다.

이렇듯 어떻게 알고리즘을 사용하느냐에 따라 프로그램의 성능을 좌우할 수 있는 것이다.

알고리즘 과목의 출제 비중

가장 최근 117회 기출문제를 통해 정보관리 종목과 컴퓨터시스템응용 종목에서의 컴퓨터 구조 과목 문제 출제 비중을 보면 다음과 같다. 알고리즘 과목은 정보관리 종목에서 출제 비중이 높은 편이다. 정보관리 종목은 프로그램 개발과 관련된 과목들이 출제 비중이 높다 보니 아무래도 알고리즘을 묻는 문제들이 자주 출제되곤 한다.





정보관리 종목 117회 영역별 출제 문항 수

영역	1교시	2교시	3교시	4교시	계
소프트웨어 공학	1	2	1	2	6
데이터베이스	2	1	0	1	4
경영정보	0	0	0	0	0
컴퓨터 구조	1	1	1	1	4
네트워크	1	0	1	0	1
정보보안	2	0	1	1	4
디지털 신서비스	4	2	2	0	8
알고리즘	2	0	1	1	4
계	13	6	6	6	31

컴퓨터시스템응용 종목 117회 영역별 출제 문항 수

영역	1교시	2교시	3교시	4교시	계
소프트웨어 공학	2	1	1	2	6
데이터베이스	0	0	0	0	0
경영정보	0	0	0	0	0
컴퓨터 구조	3	1	0	0	4
네트워크	3	1	2	1	7
정보보안	1	1	0	0	2
디지털 신서비스	4	2	3	3	12
알고리즘	0	0	0	0	0
계	13	6	6	6	31

기출문제를 통한 실전 알고리즘 문제

117회 기출문제를 통해 알고리즘 과목의 출제 경향을 알아보자.

회차	105	107	108	110	111	113	114	116	117	계	비중
정보관리	4	2	3	0	5	3	5	2	4	28	10%
컴시응	0	0	1	1	1	1	1	5	0	9	3.5%

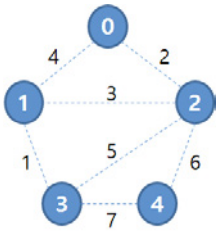
먼저, 최근 5개년 알고리즘 과목의 출제 경향을 보면 다음과 같다.

컴퓨터시스템응용 종목의 경우 1문제 나오거나 거의 출제가 되지 않는다. 그러나 116회의 경우 5문제나 출제가 되었다. 굉장히 이례적이라 볼 수 있다. 정보관리 117회와 컴퓨터시스템응용의 116회를 통해 어떤 문제들이 출제되었는지

정보관리 117회

회차	교시	문제
117회	1	4. 통계적 가설검정의 1차 오류와 2차 오류
		5. XAI(eXplainable AI)
	3	3. MST(Minimum Spanning Tree)를 구하는 알고리즘인 크루스칼(Kruskal) 알고리즘, 프림(Prim) 알고리즘을 설명하시오.
	4	6. 머신러닝(Machine Learning)과 인공지능(Artificial Intelligence)에서 편향(biased)된 결과를 만들어내는 이유를 3가지만 설명하시오.

컴퓨터시스템응용 116회

회차	교시	문제
116회	1	1. 알고리즘의 시간복잡도(Time Complexity) O(1), O(n), O(n3)에 대하여 설명하시오.
		3. 플렌옵티 영상처리(Plenoptic Image Processing)
		12. 분할 정복(Divide and Conquer), 탐욕법(Greedy), 동적계획법(Dynamic Programming)에 대하여 설명하시오.
	4	2. 아래 그래프에서 최소신장트리(MST : Minimum Spanning Tree)를 구하는 과정을 2개의 알고리즘을 이용하여 설명하시오.  가. 크루스컬(Kruskal) 알고리즘 나. 프림(Prim) 알고리즘
		6. MPEG(Moving Picture Experts Group)의 동영상 압축기법에 대하여 공간적인 측면과 시간적인 측면으로 나누어 설명하시오.

살펴보자.

출제 문제를 보면 컴퓨터시스템응용 종목의 4교시 2번 문제와 정보관리 종목의 117회 3교시 3번 문제가 교차출제 된 것을 알 수 있다. 다시 한번 기출문제의 중요성을 확인하는 시간이다. 교차출제가 될 수 있으므로 우리는 최소 5개년의 기출문제를 살펴봐야 한다.

알고리즘 과목 어떻게 준비해야 할까?

- 절대로 어려운 문제가 나오지 않는다. 기본개념에 충실하자!
  - 알고리즘은 정답이 있는 문제다. 이런 문제는 정확히 기술하지 못하면 오답이다! 정확히 기술할 수 있도록 원리를 학습해 두자.
  - 대부분이 정확히 알고 기술하지 못하므로 정확히 작성만 한다면 기본점수는 받는다.
    추가적인 고득점을 위해서는 정답 이외에 차별화하여 관련 알고리즘 및 활용방안 또는 해당 알고리즘의 단점 및 보완점까지 기술해주면 OK!
- 정보통신기술사 또는 정보관리, 컴퓨터시스템응용기술사 각각 교차 출제 비율이 높으니 반드시 다른 종목의 기출문제도 함께 학습해두자!
  - 자료구조의 기본인 정렬 및 탐색과 관련된 알고리즘은 원리까지 확실히 해두고, 알고리즘 성능평가 방법에 대한 내용도 꼭 학습하자.

• 빅데이터, 인공지능과 관련된 알고리즘을 준비하자!

- 가장 이슈화되는 부분이며 알고리즘의 중요성을 알리는 토픽이기도 하다. 최신동향에 맞춰 알고리즘을 학습해두자.

알고리즘 학습에 도움이 되는 사이트와 앱

사이트

'visualgo'(visualgo.net/ko)는 정렬, 트리 등의 원리를 이해하기 쉽게 시각화하여 한 단계씩 나타낸 사이트이다. 그림과 코드가 한 화면에 나오고, 재생 버튼을 누르면 순차적으로 for 반복문을 따라 알고리즘 진행 과정을 보여준다.

앱

앱으로 완벽히 공부할 수는 없지만, 접근성이 좋은 것은 사실이다. 개인적으로 간단하고 쉬워서 도움이 많이 된 앱은 **'알고리즘 도감'**이라는 앱이다. 위에 설명한 visualgo와 비슷하게 한 단계씩 천천히 알고리즘을 설명해주고, 간단한 예제도 지원해서 입문자에게 추천하고 싶은 앱이다.

알고리즘에 대한 학습은 데이터베이스를 운영하거나 개발프로젝트 수행 시 또는 소프트웨어 아키텍처를 설계할 때에도 큰 도움이 될 수 있다. 방송기술인들도 기술사 수험뿐만 아니라 기본적인 자료구조나 알고리즘에 대해 관심을 가져두면 업무수행에도 큰 도움이 된다.

다음 호에서는 정보시스템감리 과목에 대해 알아보겠다. 종목별 정보시스템 감리 과목의 출제 비중과 실전에서 어떤 문제들이 출제되며 또, 고득점을 위해서는 어떤 형식으로 답안을 기술해야 하는지에 대해 알아보도록 하겠다.

참고문헌

알고리즘 편 (librewiki.net/wiki), visualgo (visualgo.net/ko), blog.yena.io/studynote/2018/11/14/Algorithm-Basic.html