

# 파이썬으로 시작하는 자료구조와 알고리즘

---

## <6주차>

### WARNING

본 교육 콘텐츠의 지식재산권은 재단법인 네이버커넥트 및 노씨데브에 귀속됩니다. 본 콘텐츠를 어떠한 경로로도 외부로 유출 및 수정하는 행위를 엄격히 금합니다. 다만, 비영리적 교육 및 연구활동에 한정되어 사용할 수 있으나 양사의 허락을 받아야 합니다. 이를 위반하는 경우, 관련 법률에 따라 책임을 질 수 있습니다.

# 6 - 알고리즘 활용 (AI, 코딩테스트)

[AI 와 알고리즘](#)

[머신러닝과 자료구조](#)

[머신러닝과 알고리즘](#)

[빅데이터](#)

[자료구조 알고리즘이 필수인 이유](#)

[알고리즘 코딩테스트](#)

[코딩테스트](#)

[언어 선택](#)

[공부해야할 내용들](#)

## AI 와 알고리즘

AI는 데이터를 기반으로 패턴을 학습하고 의사결정을 자동화하는 시스템입니다. AI 시스템의 효율성과 정확성은 데이터를 처리하고 분석하는 방법, 즉 **자료구조와 알고리즘**에 크게 의존합니다.

- AI 모델은 **입력 데이터**를 받아 **출력 결과**를 생성합니다.
- 이 과정에서 효율적인 자료구조와 알고리즘이 필수적입니다.
  - 데이터 전처리: 효율적인 탐색과 필터링.
  - 모델 학습: 최적화 알고리즘과 데이터 관리.
  - 예측/추론: 빠른 응답 시간과 메모리 효율성.

## 머신러닝과 자료구조

- 머신러닝은 **데이터를 학습**하여 예측 모델을 생성합니다.
- 자료구조 활용 예시
  - **배열**: 데이터를 배치하고 반복적으로 처리.  
ex) numpy 배열을 사용한 벡터 연산.
  - **해시테이블**: 데이터의 중복 제거와 빠른 검색.  
ex) 특성 벡터에서 유니크한 값을 관리.
  - **트리**: 결정 트리 기반 알고리즘.  
ex) 랜덤 포레스트, 그래디언트 부스팅.

## 머신러닝과 알고리즘

- 머신러닝 모델 학습에서 **최적화 알고리즘**이 중요합니다.
  - 경사 하강법: 최적의 모델 파라미터를 학습.
  - 동적 프로그래밍: 반복적 계산을 줄여 효율성 증가.

- 탐욕 알고리즘: 부분적으로 최선의 선택을 반복.

## 빅데이터

- 빅데이터는 대량의 데이터를 처리하기 위한 효율적인 관리가 필요합니다.
  - **그래프**: 소셜 네트워크 분석.  
ex) 사용자 간 관계를 그래프로 모델링.
  - **트라이**: 텍스트 데이터의 패턴 분석.  
ex) 자동완성 시스템.
  - **큐**: 스트림 데이터 처리.  
ex) 실시간 로그 분석.

## 자료구조 알고리즘이 필수인 이유

- **성능 향상**
  - AI와 빅데이터는 대량의 데이터를 처리합니다.
  - 비효율적인 알고리즘은 처리 속도를 느리게 하고, 메모리를 낭비합니다.
- **확장성**
  - 효율적인 자료구조와 알고리즘은 대규모 데이터와 복잡한 모델에서도 안정적으로 동작합니다.
- **정확도**
  - 일부 AI 문제는 탐색 알고리즘을 기반으로 최적의 해를 찾아야 합니다.
  - 정확한 알고리즘 선택이 결과의 품질에 직접적으로 영향을 줍니다.

## 알고리즘 코딩테스트

개발자 취업은 통상 다음과 같은 절차로 진행됩니다.



**서류 제출 → 알고리즘 코딩 테스트(또는 과제형 코딩테스트) → 기술면접 → 임원면접**

회사에 따라 코딩테스트의 유형이 다양합니다.

- 코딩테스트 **✗** : 규모가 작은 회사 또는 경력이직
- 실무과제형 코딩테스트: **스타트업, 상시채용, 경력이직**  
여러 회사들이 실험적으로 도입하고 있습니다. 웹개발자 채용의 경우 정해진 기간동안 주어진 명세에 따라 웹 개발을 완료하면 됩니다.
- 알고리즘 코딩테스트 : **대기업, 대규모 채용, 신입, 1~3년차 이직**  
많은 IT회사에서 알고리즘 코딩테스트를 봅니다. 특히 카카오, 네이버, 라인, 쿠팡, 삼성, SK 등등 개발자들이 가고

싶어하는 높은 연봉의 IT 기업에서는 거의 필수로 봅니다. 따라서 알고리즘 코딩테스트를 준비해 둔다면 지원할 수 있는 회사의 폭이 굉장히 넓어집니다.

## 코딩테스트

- 코딩테스트 목적(회사 입장): 지원자의 **구현능력** 과 **문제 해결능력** 평가
  - 구현능력 : 기본적인 언어 문법, 자료구조, 알고리즘, 문자열 다루는 능력
    - 자료구조, 알고리즘등은 수업에서 배운 것들을 반복 학습만 하면 정복 가능하다.
    - 템플릿처럼 제공하는 코드들을 이해한 후 나만의 코드로 만들어보고 외우자.
  - 문제 해결 능력: 문제 이해
    - 시뮬레이션 문제들
    - 점점 문제 자체를 이해하는게 어려워지고 있다.
    - 문해력이 중요하다.
    - 문제해결능력은 단기간에 실력을 올리기 쉽지 않다.
    - 못풀겠는 문제를 스스로 해결하고자 하는 노력을 많이 해봐야 조금씩 는다.
- 코딩테스트 목적(취준생 입장): 1등이 아니라 PASS가 목표다.
  - 빈도 높은 순서로 공부하자.
  - 정확성 테스트부터 통과하는게 목표다. 효율성테스트(시간복잡도)는 나중에 걱정하자.
- 자주 나오는 자료구조 / 알고리즘
  - 빈도 높음
    - BFS, DFS, 완전탐색, 시뮬레이션(카카오 코딩테스트, 삼성 코딩테스트), 문자열, 해시맵, 정렬
    - 시뮬레이션
      1. 자물쇠와 열쇠 <https://school.programmers.co.kr/learn/courses/30/lessons/60059>
      2. 뱀 <https://www.acmicpc.net/problem/3190>
      3. 기둥과 보 설치 <https://school.programmers.co.kr/learn/courses/30/lessons/60061>
      4. 치킨 배달 <https://www.acmicpc.net/problem/15686>
      5. 외벽점검 <https://school.programmers.co.kr/learn/courses/30/lessons/60062>
  - 빈도 중간
    - 스택, 큐, 다익스트라, 우선순위 큐, 그래프, 순열 조합, DP, two pointer(sliding window)
  - 빈도 낮음
    - 위상 정렬, 백트래킹, 트리, 링크드 리스트

## 언어 선택



## 코딩테스트에 유리한 언어

- python을 가장 추천.
  - 표준라이브러리 강력
  - 문자열 처리 강력
  - 배우는데 걸리는 시간이 굉장히 짧다.
  - C++보다 실행속도가 느리다고 하지만, 기업용 코테에서는 실행속도보단 구현과정의 시간 단축이 훨씬 중요하다.
- 입사하려는 회사에서 요구하는 언어가 있다면 해당 언어로 구현할 수 있게 연습해놓자.
  - ex) 백엔드 자바 개발자 - JAVA,
  - ex) 프론트엔드 react 개발자 - JS

언어 선택은 굉장히 많이 묻는 질문이지만, 쉽게 답하기가 어렵습니다.

일단 제가 JAVA 개발자라면 몇몇 회사에서 JAVA로만 코딩테스트를 보기를 요구할 수 있습니다. 그러니 결국 JAVA로 코딩테스트를 연습해봐야 합니다. JS 프론트엔드 개발자나 Swift 개발자도 똑같습니다. (C/C++ 개발자는 보통 C/C++로 시험을 치는 걸 선호하니 넘어가고, python 개발자는 python으로 하면 됩니다.)

그렇다고 JAVA, JS, Swift로 코딩테스트를 공부하면 구현 자체가 어렵다 보니 자료구조/알고리즘/문제 해결 능력을 학습할 시간이 줄어듭니다. 배보다 배꼽이 더 커지게 됩니다.

그래서 일반적으로 권장하는 방법은 Python을 사용하여 자료구조, 알고리즘, 문제 해결 능력을 향상시키고, Python으로 문제를 풀면서 적응 능력을 강화하는 것입니다. 그 후, 시간을 따로 내어 Java, JS, Swift 등으로 구현 연습을 하는 것이 좋습니다.

시간이 없어서 조금하게 느껴진다면 어쩔 수 없이 조금 포기할것들이 생깁니다.

1. python을 선택해서, JAVA(JS,Swift) 로만 코딩테스트를 봐야하는 경우는 지원하지 않는다.
2. 처음부터 JAVA(JS,Swift)로 코딩테스트를 공부한다 → 이 경우 코딩테스트 학습속도를 포기해야합니다.

잘 생각해보고 자신의 상황에 맞게 언어를 선택하시기 바랍니다.

## 공부해야할 내용들

### 1. 자료구조 & 알고리즘

대학교에서 배우는 자료구조와 알고리즘, 그리고 시중에 있는 서적이나 강의를 듣고나서 코딩테스트 문제를 풀어보 시면 상당한 괴리감이 느껴질 것입니다. 면접에서 물어볼만한 것들과 코딩테스트에서 나올만한 내용이랑 차이가 있기 때문입니다.

예를 들면 면접에서는 Sorting에 대한 부분들이 많이 나오지만 코딩테스트에서는 정렬 구현 능력이 전혀 필요 없습니다. 언어에 내장되어있는 정렬 함수를 사용하면 됩니다. 또한 해시테이블의 collision에 대한 대처 방법들을 면접에서는 굉장히 중요하게 다룹니다. 하지만 코딩테스트에서는 어떤 상황에서 해시테이블을 사용하면 시간복잡도

를 줄일 수 있는지를 판별하는 능력이 중요합니다. 언어별로 해시테이블은 이미 깔끔하게 정의 되어 있기 때문에 구현할 필요가 없기 때문입니다.

알고리즘 코딩테스트의 목적은 어렵고 복잡한 알고리즘을 많이 알고 있는지 물어보는 것이 아닙니다. 간단하지만 핵심적인 알고리즘들을 제대로 이해하고 있고, 문제를 잘 해석하여 상황에 맞는 알고리즘을 제대로 사용할 줄 아는지를 평가하는게 주된 목표입니다.

### 이론과 실전의 괴리

✓ 자료구조의 직접적인 구현보단, 주어진 문제에 적합한 자료구조를 선택하는 능력이 필요하다.

✓ 언어별로 잘 정의된 함수를 잘 사용해야 한다.

ex) 정렬 알고리즘 종류별로 구현을 못해도 된다. 잘 정의된 정렬함수를 이용할 줄만 알면 된다.

✓ 굉장히 복잡한 알고리즘을 묻는 문제는 별로 없다. 나온다고 해도, 그 문제를 못풀어도 합격할 수 있다.

✓ 시간복잡도, 문자열처리, 구현 능력등이 생각보다 훨씬 중요하다.

✓ 재귀에 대한 이해가 큰 차이를 만든다.

## 2. 구현 능력 (언어 숙련도)

코딩테스트는 결국 구현을 제대로 해야 합격을 할 수 있습니다. 머릿속에 아무리 좋은 알고리즘이 떠올라도 그것을 구현하지 못하면 망입니다. 특히 요새는 검색도 안되고 IDE 사용도 안되는 시험 환경인 경우가 많습니다. 따라서 코딩테스트에서 사용할 주력 언어를 하나 선택하고 문법에 익숙해져야 합니다. 하지만 언어를 먼저 마스터하고 넘어가야겠다는 생각은 버려야 합니다. 반복문과 조건문, 변수선언 등 기본적인 문법만 익힌 후 자료구조와 알고리즘을 익히면서, 그리고 문제를 풀면서 모르는걸 그때 그때 찾아보며 공부하시는걸 추천드립니다.

구현 능력을 키우는 가장 확실한 방법은 많이 풀어보는 것입니다. 풀었던 문제를 먼저 5~10번 복습을 하면서 술술 코드가 나올정도로 익숙해져야 합니다. 그리고나서 새로운 문제를 풀면서 머릿속에서 생각난 코드 구현

### 언어선택 가이드

코딩테스트 문제를 잘 풀기 위해서는 기본적으로 코드 구현력이 필요합니다. 코드 구현력이란 내가 생각한 알고리즘을 코드로 구현할 수 있는 능력을 말합니다. 따라서 가장 자신 있는 언어를 선택하는 것이 좋습니다. 만약 특출나게 자신있는 언어가 없다면 학습기간이 짧으면서 코딩테스트에 유리한 python을 추천합니다. Python과 C++의 다양한 라이브러리는 코딩테스트 문제를 풀 때 강력한 도구가 됩니다.

## 3. 문제 풀기

수학 이론을 완벽히 이해한 것 같아 보여도 막상 문제를 풀어보면 틀렸던 경험들이 있을겁니다. 운동도 이론을 제대로 알고 있어도 실제로 해보지 않으면 근육을 제대로 쓸 수 없습니다. 코딩테스트도 동일합니다. 이론을 제대로 익혀도 문제를 많이 풀어봐야 실력이 늘어납니다.

그렇다고 무턱대고 문제부터 푸는 행동은 시간을 낭비하는 지름길 입니다. 효율적으로 실력이 늘기 위해서는 제대로 배운 이론을 바탕으로 문제를 풀면서 틀려도 보고 실수도 하면서 적용해보는 과정이 필수적입니다.

본 강의에서 몇몇 문제를 같이 해결해 볼 것입니다. 코드를 작성하지 않고, 어떤식으로 문제를 해결해나갈지 아이디어를 떠올리는 과정을 알려드릴 것입니다. 강의에서 배우는 문제에 대해서 복습을 5~10번 하여 완전히 정복하게 중요합니다. 이것이 기본기가 되어 추후에 새로운 문제에서도 응용할 수 있습니다.

결국 코딩테스트 문제는 비슷한 유형일 수 밖에 없습니다. 복습이 충분히 되었다면 아래 사이트에서 문제를 풀어보길 추천합니다.

---

📌 문제풀기 좋은 사이트

- 삼성 코딩테스트 기출 🖱️ [백준 삼성 SW 역량 테스트 기출 문제](#)
  - 카카오 코딩테스트 기출 🖱️ [프로그래머스 카카오 BLIND채용 기출 문제](#)
  - 알고리즘 별로 연습하기 좋은 문제 모음 🖱️ [프로그래머스 코딩테스트 고득점 Kit](#)
  - 해외 유명한 문제 연습 🖱️ [Leetcode](#)
-