

AI 시대의 SW개발자 학습전략

한국공학대학교 컴퓨터공학부
전 광 일

소프트웨어 개발자 학습 목표

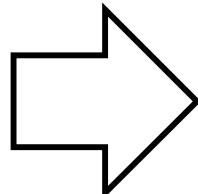
- ❖ 기본을 충실히 다져라. (Back to the Basic)
- ❖ 변화에 적응하는 자기주도적 학습 (Self Directed Learning)
- ❖ 문제 정의 및 해결 역량(Problem Defining and Problem Solving Capability)
 - 충실한 기본 지식을 활용

SW 개발자 학습 방법의 변화

❖ 3년 간의 학습(2학년 ~ 4학년)

❖ AI 시대 이전의 학습

What (기본/핵심 개념 이해)	Why (응용분야, 장단점)
How-to (구현방법)	



❖ AI 시대의 학습

What (개념 이해)	Why (응용분야, 장단점)
AI 활용 역량	

SW 개발자 학습 방법의 변화

❖ 질문 1

- 100만 건의 데이터가 있을 때 100만 건의 트랜잭션의 구성이 searching 99%, updating이 1%로 구성된 경우 가장 좋은 자료구조는 무엇인가?

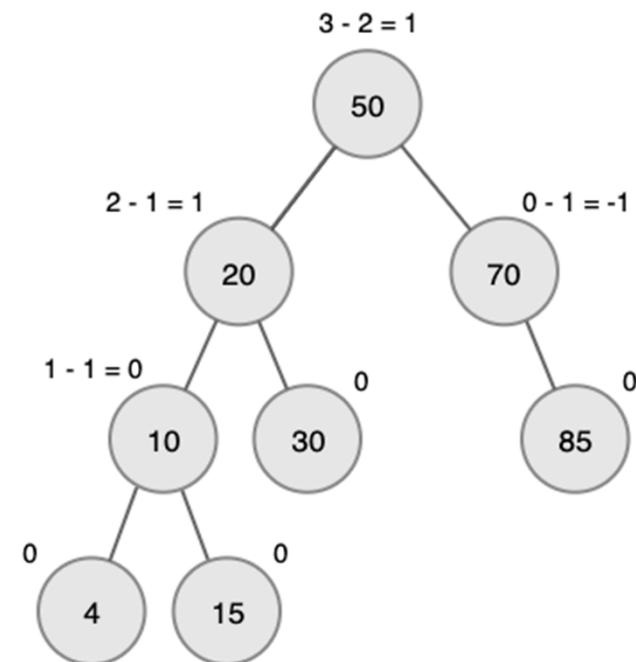
❖ 질문 2

- 100만 건의 데이터가 있을 때 100만 건의 트랜잭션의 구성이 searching 70%, updating이 30%로 구성된 경우 가장 좋은 자료구조는 무엇인가?

핵심 개념의 이해

❖ AVL(Adelson-Velsky and Landis) Tree

- 모든 노드에 대해서 각각의 왼쪽 서브트리와 오른쪽 서브트리의 높이 차이가 1이하인 트리
- 장점: 가장 빠른 탐색 성능
- 단점: 삽입/삭제 시 높은 오버헤드
- 참고:
<https://velog.io/@dankj1991/Tree-AVL-Tree>



핵심 개념의 이해

❖ Self-Balanced Binary Search Tree

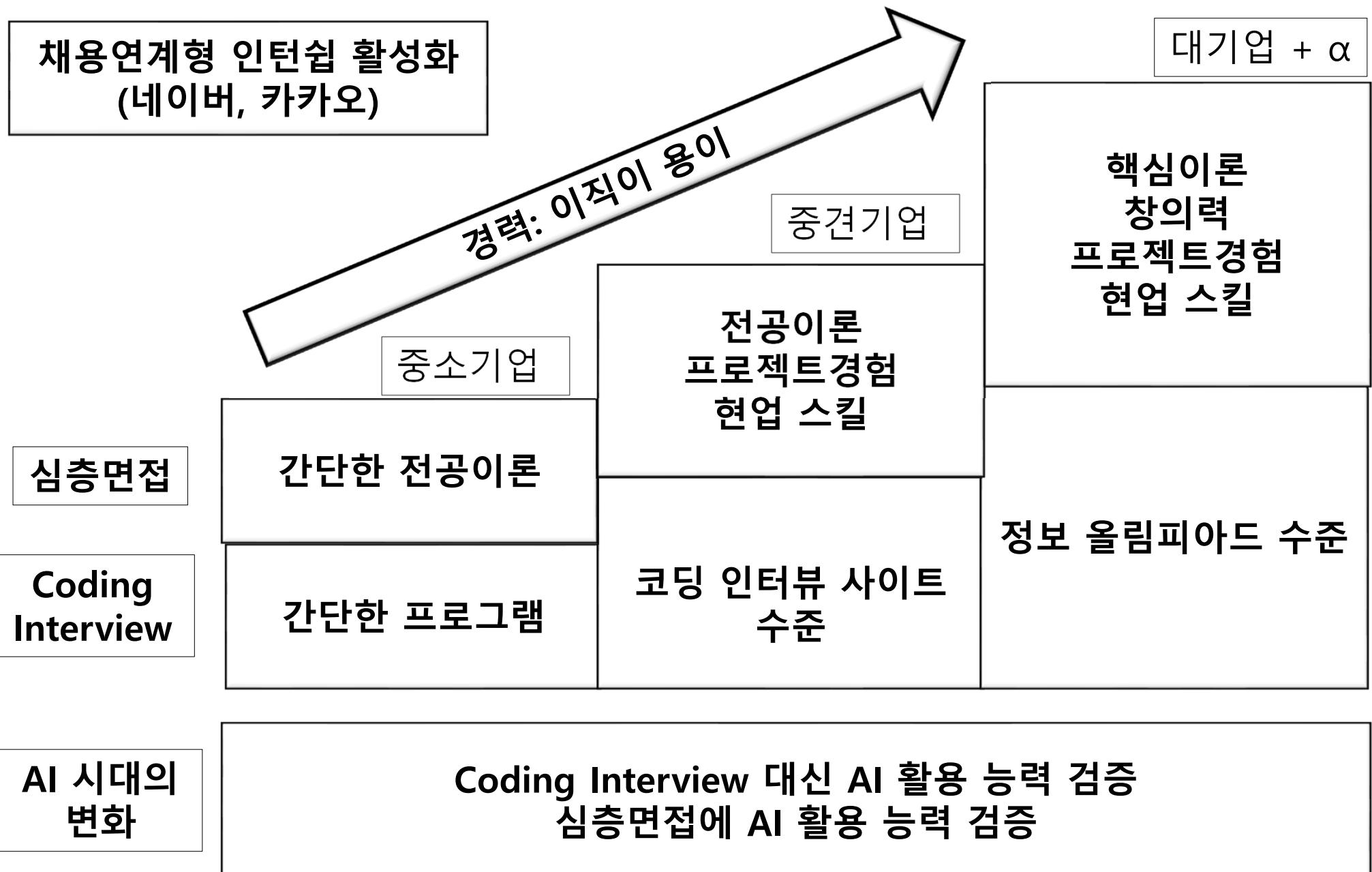
종류	주요 특징	장점	단점
AVL 트리	모든 노드의 좌우 서브트리 높이 차를 1 이하로 엄격하게 제한	탐색 속도가 가장 빠름 (균형이 매우 정교함)	삽입/삭제 시 균형을 맞추기 위한 회전(Rotation)이 잦음
Red-Black 트리	노드에 색상(적/흑)을 부여하여 느슨하게 균형을 유지 (C++ STL, Java TreeMap 등에서 사용)	삽입/삭제 작업이 AVL보다 빠름 (실무에서 가장 널리 사용됨)	AVL 트리보다 트리의 높이가 높아 탐색 속도는 약간 느릴 수 있음
B-트리	하나의 노드에 여러 데이터를 저장. 자식 노드 수가 2개 이상인 다원 트리	대량의 데이터를 처리할 때 디스크 접근 횟수를 최소화함 (데이터베이스, 파일 시스템에 필수적)	구조가 복잡하며, 데이터가 적을 때는 메모리 낭비가 발생할 수 있음
B+트리	B-트리의 변형으로, 모든 데이터는 리프 노드에만 저장하고 연결 리스트로 연결	범위 탐색(Range Scan)과 전체 순회에 매우 효율적 (현대 DB 인덱스의 표준)	B-트리와 마찬가지로 구현이 복잡함
Splay 트리	최근에 접근한 노드를 루트로 옮기는 방식 (Move-to-root)	자주 쓰이는 데이터에 매우 빠르게 접근 가능 (캐싱 효과)	특정 상황에서 트리가 한쪽으로 치우칠 수 있어 최악의 경우 성능 저하 위험이 있음
Treap (트립)	이진 탐색 트리와 힙(Heap)의 특성을 결합 (노드에 무작위 우선순위 부여)	확률적으로 균형이 잡히며 구현이 상대적으로 매우 간단함	확률에 의존하므로 이론적인 최악의 성능이 나타날 가능성성이 미세하게 존재함

AI 활용 역량(googling 포함)

- ❖ 구체적이고 명확한 질의 문장(prompt) 작성 능력
 - 핵심 개념과 응용 분야에 대한 이해가 중요
- ❖ 결과 검증의 중요성
 - AI가 항상 정확한 코드를 생성하지 않는다. (기본적인 코딩 능력 필요)
 - 기술 부채(Technical Debt)
- ❖ 반복적인 정제: AI와의 피드백 반복

- ❖ 학습 전략
 - 3학년까지 핵심 개념과 응용분야에 대한 정리에 생성형 AI 활용
 - 4학년 때 생성형 AI도구 적극 활용

AI 시대의 채용 패턴의 변화



SW 개발자 취업 시장

- ❖ SW 개발자 직업은 사라지지 않을 것이다.

- 역할의 변화(Coder에서 Architect로)
- 전체 수요는 줄어들 것 같다. (30% 내외 ?)

- ❖ 신입(주니어) 개발자 채용

- 신입 개발자의 채용은 계속 될 것이다. (기업의 미래)
- AI 활용 역량을 갖춘 주니어개발자의 등장

- ❖ 1인 개발자 시대(창업)

- 회사경험 5년 이상 추천(창업 아이템 발굴, 인맥, fund 조성 등에 유리)

컴퓨터/소프트웨어 직장 비교

	자체개발조직을 갖춘 서비스 기업	SI (System Integration) / 기업용 SW 제공	SW 융합 제조 (임베디드시스템)	기타
정의	<ul style="list-style-type: none"> ■ 서비스를 제공하는 기업이 자체적인 개발 조직을 운영 ■ 주로 앱 개발자 	<ul style="list-style-type: none"> ■ 서비스를 제공하는 기업의 요청을 받아 서비스를 개발해주는 SI기업/기업용 SW 제공기업 	<ul style="list-style-type: none"> ■ SW 기술을 이용하여 제품을 생산하여 소비자에게 판매하는 기업 	<ul style="list-style-type: none"> ■ 보안 업체 ■ 공공기관 및 일반 기업의 IT
대표적인 기업	네이버, 카카오, 라인 쿠팡, 배달의 민족, 당근마켓, 토스뱅크, 지그재그, ...	삼성SDS, LG CNS, SK C&C, 한화시스템, 롯데정보통신, 신세계 I&C, ... AWS, Microsoft, Google, SAP, IBM, salesforce, ... Partner사로 불리는 중견업체	삼성전자, LG전자, SK 하이닉스, 현대자동차, ... 기타 중견 제조업체 반도체장비, 전장	정부/공공기관 IT센터 은행, 증권, 대기업/중견기업의 IT센터
특성	<ul style="list-style-type: none"> ■ 회사의 자체 서비스를 개발하므로 팀 간의 평등한 의사 소통 ■ 프로젝트 수행 시 업무 강도가 낮을 가능성 	<ul style="list-style-type: none"> ■ 수주를 준 업체와 SI업체 간의 갑을 관계가 형성 ■ 프로젝트 수행 시 업무강도가 높을 가능성 ■ 기업용 SW 독과점 기업 (외국계 기업이 많음) 	<ul style="list-style-type: none"> ■ 자체적으로 SW를 개발하거나 용역을 줌 ■ 프로젝트 수행 시 업무 강도가 낮을 가능성 	<ul style="list-style-type: none"> ■ 회사의 IT 자산 서비스 기획, 운영 관리
RISK	<ul style="list-style-type: none"> ■ 소속회사의 수익에 민감(회사가 어려워지면 정리하고 가능성 높음) 	<ul style="list-style-type: none"> ■ SI 수주해서 프로젝트를 진행함으로 다양한 일감이 존재함 	<ul style="list-style-type: none"> ■ 회사의 수익에 민감하나 일반적으로 안정적 	<ul style="list-style-type: none"> ■ 안정적이거나 취약성이 난이도가 높음

❖ SW 개발 역량과 더불어 비즈니스 도메인 지식이 매우 중요

취업 전략 추천

- ❖ 4학년 여름방학(2학기)
부터 구직 활동 시작
 - 역량이 높은 학생: 4학년 2학기 ~ 졸업 후 상반기 (5월): 본인이 원하는 기업에 지원해보고 안되면 중견/중소기업 지원
 - 역량이 낮은 학생: 4학년 2학기부터 중견/중소기업 지원
- ❖ SW개발자 직업 진입이 가장 중요
 - 자기소개서와 포트폴리오의 내용이 매우 중요



참고 문헌

- ❖ 에디 오스마니 저, 강민역 역, 바이브 코딩 너머 개발자 생존법, 한빛미디어, 2025. 11
- ❖ 최지호(코드팩토리) 저, 클로드 코드 완벽가이드, 골든 래빗(주), 2025. 9
- ❖ 조태호 저, 혼자 공부하는 바이브 코딩 with 클로드 코드, 한빛미디어, 2025. 12.