

# KAIST 전산학부 입시학습 자료

전산학 주요 교과목에 대한 석사과정 구술면접 준비를 위한 학습 보조자료입니다. 각 과목의 핵심 키워드와 함께 구술면접에서 나올 가능성이 있는 질문들을 교과목별로 정리하였습니다. 2014년에 공개된 자료를 바탕으로 2024년에 전산학부 전임교원의 피드백을 받아 업데이트하였습니다. 본 자료는 전산학부 기초과목 입시 준비를 위한 학습 보조자료로만 활용하시기 바랍니다. 아래 내용을 참고하여 교과서 내용을 리뷰를 체계적으로 하는데 도움이 될 것으로 기대됩니다.

전산학부 입시위원회

2024.8.1

|                                      |    |
|--------------------------------------|----|
| Programming Language / Compiler..... | 3  |
| Architecture.....                    | 6  |
| OS.....                              | 9  |
| DB.....                              | 12 |
| 이산 수학, 확률, 통계.....                   | 15 |
| 네트워크.....                            | 18 |
| Software Engineering.....            | 21 |
| AI / ML.....                         | 24 |
| 알고리즘.....                            | 26 |
| 그래픽.....                             | 29 |
| 정보 보호.....                           | 31 |

## Change Log

- 26/4.27 AI/ML & HCI & AI updated by Uichin Lee

# Programming Language / Compiler

## 〈Keywords〉

- Syntax and Semantics
- Identifiers
- First-order Function, First-Class Function, Recursion
- Mutable Data Structure, Variables
- Lazy Evaluation
- Continuations, Low-Level Representation of Interpreters
- Memory, Garbage Collection
- Types, Type Checking, Typed Recursion
- Variants
- Type Inference
- Parametric Polymorphism, Subtype Polymorphism
- Type Soundness

## 〈Preliminaries〉

- 자바에서 public, protected, private 키워드가 있는데 아무것도 안 쓸 경우 default로 적용되는 범위는? 그렇게 디자인된 이유는?
- C++에서 서로 다른 타입의 오브젝트를 가리키는 포인터를 사용할 수 있나?
- 전역 변수 사용의 장단점은?

## 〈Function〉

- Higher-order function이 무엇인지 설명하고, 현대 프로그래밍에서 이 개념을 활용하여 쓰이는 대표적인 함수를 제시하시오.

## 〈Lazy Evaluation〉

- 파라미터 패싱 방식에는 eager evaluation 방식과 lazy evaluation 방식이 있다. 두 방식의 차이점을 비교 설명하세요. call-by-value와 call-by-name 파라미터 패싱 방식은 각 각 어느 방식에 속하는지 구분하세요.

## 〈Memory, Garbage Collection〉

- Rust와 Java는 C/C++를 대체하여 시스템 프로그래밍을 쉽게 하기 위해 대두된 대표적인 언어들이다. 메모리 관리 측면에서 기존 C/C++의 한계를 제시하고, 두 언어가 그 한계를 각각 어떻게 극복했는지 설명하고, 어떤 장단점이 있는지 설명하시오.

- RAII(Resource acquisition is initialization)가 무엇이고, 어느 상황에서 사용하면 좋을지 설명하시오.
- 가비지 컬렉션을 위해 Mark and Sweep 기법이 크게 어떻게 이루어지는지 간단히 설명하고, 이 기법의 장단점을 설명하시오.

### <Type System>

- Type system이 sound하다는 것, 그리고 complete하다는 것의 의미를 각각 설명하시오.
- 프로그램 실행 전 컴파일 시에 타입을 정적으로 검사하는 프로그래밍 언어가 그렇지 않은 프로그래밍 언어에 비해 갖는 장점을 설명하시오. 또한, 그럼에도 불구하고 그렇지 않은 언어가 사용되는 이유를 설명하시오.
- Static typing과 dynamic typing의 차이점은? C++/JAVA은 어떤 typing을 사용하는가?
- Option type이 무엇이고, 어느 상황에서 사용하면 좋을지 설명하시오.

### <Compiler>

- 중간 표현(Intermediate Representation, IR)은 컴파일러 설계에서 매우 중요한 요소이다. IR이 사용되는 이유를 간단히 설명하라.

### <기타>

- 아래 두 Javascript 코드(code1.js 및 code2.js)의 실행 결과가 다른 원인을 설명하고, 그 원인 각각의 장단점을 설명하시오.

|                                                                                                                                                                    |                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>// code1.js let l1 = [1, 2]; let l2 = [100, 200];  let s = 0; for (var x of l1) {   for (var x of l2) {     s += x;   } } console.log(s) // Result: 400</pre> | <pre>// code2.js let l1 = [1, 2]; let l2 = [100, 200];  let s = 0; for (let x of l1) {   for (let x of l2) {     s += x;   } } console.log(s) // Result: 3</pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- 간단한 표현식  $1 + 2 * 3$  을 중심으로, Operational semantics와 Denotational semantics의 차이를 설명하시오.
- 아래의 코드에서 sum 변수에 배열 array의 모든 요소를 더하는 작업을 수행한다. 컴파일러가 이 루프를 최적화할 수 있는 방법을 제안하고, 이 방법이 성능 향상에 어떻게 기여하는지 설명하라.

```
int sum = 0;
for (int i = 0; i < 1000; i++) {
    sum += array[i];
}
```

- 다음은 프로시저  $read(a, b)$ 의 의사코드이다.

```
c := 3.0 * a + b;
if c = 0 { a := 1; } else { a := 1.0/c + 1.0/b; }
```

이 프로그램은 특정 조건을 만족하는  $a$ 값과  $b$ 값이 들어오면 오류를 일으킨다. 다음 중 프로그램의 오류를 막을 수 있는 가장 약한 조건은?

- 1)  $b > 0$
- 2)  $a > 0$  and  $b > 0$
- 3)  $a \neq -b/3$
- 4)  $b \neq 0$
- 5)  $3.0*a \neq 0$  and  $b \neq 0$

# Architecture

## 〈Keywords〉

- ISA (Instruction set architecture)
- Pipeline Architecture
- Control Hazard
- Memory Hierarchy, Cache
- Virtual Memory
- ILP (Instruction level parallelism): Superscalar, Out-of-order execution
- Multiprocessor, Cache coherence
- Multi-core Architecture
- I/O system

## 〈ISA (Instruction set architecture)〉

- CISC와 RISC의 차이점은 무엇인가?
- Register File과 Cache의 차이는 무엇인가?
- Intel과 AMD가 만드는 CPU의 공통점과 차이점은 무엇인가?

## 〈Pipelined Architecture〉

- 파이프라이닝은 왜 하는가?
- 파이프라인 해저드(Pipeline Hazard)란 무엇인가? 그리고 종류에는 무엇이 있는가?
- 파이프라이닝에서 Branch Prediction의 역할은 무엇인가?

## 〈Memory Hierarchy〉

- 캐시가 필요한 이유는? Cache hit ratio 에 대해 설명하시오.
- 메모리 접근하는데 x 사이클이 걸리고 캐시에 접근하는데 y 사이클이 걸리며 캐시 hit rate 가 h %일 때 effective access time은?
- 페이지 폴트는 언제 발생하는가? 페이지 폴트 비율과 cache miss 비율 중 큰 것은? 그 이유는?

## 〈Virtual Memory〉

- 캐시 메모리와 메인 메모리의 주소 지정 방식의 차이점이 무엇인가?

## 〈ILP (Instruction Level Parallelism)〉

- Superscalar와 Out-of-Order execution(OoO)의 모두 CPU의 성능을 높이기 위한 기법이다. 두 기법의 공통점과 차이점에 대해서 설명하시오.
- 슈퍼스칼라와 VLIW(Very Long Instruction Word)의 차이점은 무엇인가?

### <Cache>

- 캐시를 구성하는 컴포넌트에 무엇이 있는가? 각 컴포넌트는 어떤 역할을 하는가?
- 캐시에서 태그 매칭이 무엇이고 왜 필요한가?
- 캐시에서 블록(라인) 크기를 크게 했을 때와 작게 했을 때 어떤 장단점이 있을까?
- Direct-mapped cache와 set-associative cache의 장단점은 무엇인가?
- Write-through cache와 write-back cache에 대해서 설명해 보시오.
- Write-through cache는 write buffer를 보통 사용하는데 이의 역할은 무엇인가?
- Cache에서 사용하는 replacement policy에는 어떤 것이 있는가?

### <Multiprocessor, Multi-core>

- 코어가 10개 있는 cpu에서 프로세스 하나를 균등하게 처리하면 이상적인 경우, 시간이 얼마나 줄어드는가? 그런데 프로세스에서 분할되지 않는 20%의 작업이 있다고 하면, 시간이 얼마나 줄어드는가? 이제 코어가 수백개, 혹은 무한개 있다고 하면 얼마나 줄어드는가?

# OS

## <Keywords>

- Kernel
- Process, Thread, Scheduling
- Virtual Memory
- Address Translation, Paging, Swapping
- Concurrency: Locks, Conditional Variable, Semaphore
- Storage
- File System, Crash Consistency

## <Kernel>

- Kernel을 필요에 의해 어느정도 수정했다고 하자. 이 kernel이 제대로 작동하는지를 알기 위해서 어떤 test를 해야 하나?
- Interrupt란 무엇인가? interrupt를 두 종류로 나눈다면 어떻게 되는가? (software interrupt/hardware interrupt) 두 interrupt의 차이는 무엇인가? 두 interrupt의 handler를 서로 구분해서 구현해야 하는 것이 좋은가, 아니면도 상관 없는가?
- 시스템 콜과 인터럽트의 차이는? 인터럽트가 걸리면 어떤 일이 일어나고 처리 후에 어떻게 이전 상태로 돌아가는가?

## <Thread & Process>

- Thread와 process의 차이는 무엇인가?
- IPC가 무엇인가?
- Thread끼리 context switching 하는 과정에 대해 설명하시오. 같은 프로세스 내부의 thread들끼리 전환되는 것과 다른 프로세스간의 thread 끼리 전환되는 것이 어떻게 다른가?
- Non-preemptive scheduling이란?
- 요즘 많은 사람들이 각각 자신만의 스마트폰을 사용하고 있다. 이처럼 각 개인 스마트폰을 위하여 process system을 design하려고 한다. 스마트폰에서 각 application이 하나의 process로 독립해서 실행하는 것이 나을까? 아니면, 하나의 thread로 만들어져서 실행하는 것이 나을까? 어느 쪽이 나을지 결정하고, 그 이유를 설명하라.

## <Concurrency>

- Sequential program과 multithread program에서 error detection에 대한 차이점에는 어떤 것이 있나?
- Critical section이란?
- 세마포어의 개념은? 주요 연산 2가지는? 어떻게 구현해야 하는가?

### <Virtual Memory>

- Virtual Address와 Virtual memory에 대해 설명하시오.
- 어떤 C program으로 작성되어 수행중인 process가 있다고 가정하자. C언어에서는 직 접적으로 주소를 변수에게 지정해 줄 수 있다. 만약 주소 1, 2, 3, 4에 변수를 잡아서 어떤 일을 수행하는 process라고 하자. 이 process를 한 시스템에 동시에 두번 수행 시켰다. 그랬을 때 한 프로세스가 1, 2, 3, 4 주소에 있는 변수를 바꿨을 때 다른 프로세스의 변수들에도 영향을 끼치는가?
- C로 숫자로 직접 입력된 주소를 참조하는 프로그램을 짜서 컴파일한 후, 두 개를 실행시켰다고 하자. 그러면 이 두 프로세스는 물리적으로 같은 곳을 참조하나? 만약 아니라면, 어떻게 서로 다른 물리적 공간을 참조할 수 있나?
- Logical address와 physical address의 차이는 무엇인가?
- Logical address를 physical address로 바꾸어주는 hw가 무엇인가?
- 프로세스는 가상 메모리 주소로 어떻게 실제 메모리 주소를 찾아가는가?
- OS의 캐시 메모리의 사이즈를 구하기 위한 프로그램을 어떻게 구현하면 되는가?

### <Paging>

- Paging이 무엇인가? paging을 할 때 어떻게 실제 메모리 주소에 데이터를 전송하는가? page table에는 어떤 항목이 저장되는가? page table은 어디에 저장되는가? page table이 메모리에 저장되면, paging을 할 때 메모리를 두 번 참조해야 되는데, 좀 더 빠르게 하는 방법은 없는가? TLB에는 어떤 항목이 저장되는가?
- Virtual memory에서 page replacement policy에는 어떤 것이 있는가? LRU의 단점은?
- 운영 체제에서 memory management 기법중 하나로 paging이 많이 사용되고 있다. Paging 기법의 장·단점으로는 무엇이 있으며, hierarchical paging 혹은 inverted page table은 어떤 환경에서 유리한가?



# DB

## <Keywords>

- Relational Model
- SQL
- ER Model
- Database Design (Normalization)
- Complex Data Types
- Database for Big Data

## <Preliminaries>

- DBMS와 file management system의 차이점은?
- Logical database design과 physical database design의 차이점은?
- DBMS의 세 단계 data abstraction은 (physical, logical, view level)?
- Physical data independence란?

## <Relational Model>

- Database schema의 주요 구성 요소는?
- Primary key와 foreign key의 차이점은?
- Referential integrity는 무엇인가? 이를 violate하지 않도록 제공되는 기능은?
- Relational algebra의 5개 기본 연산자는?

## <SQL>

- DDL과 DML의 차이점과 예제 명령어는?
- Inner join과 outer join의 차이점은? Outer join은 왜 필요한가?
- 인덱싱 알고리즘을 하나 설명하시오.
- SQL 질의 처리시 index(예: B+-tree)를 통해 처리 속도가 빨라지는 원리는?
- Nested (sub) query는 무엇이며 언제 사용하는게 좋은가?
- SQL의 having 절과 where 절의 역할과 차이점은?

## <Database Design>

- Data inconsistency란 무엇인가?
- ER model은 무엇이며 왜 필요한가?
- Many-to-many relationship으로 연결된 두 entity set을 relational schema로 표현하기 위해 몇 개의 relation이 필요한가?
- Data normalization은 무엇이며 왜 필요한가?
- Functional dependency는 무엇인가? 간단한 예시는?
- 3NF과 BCNF의 차이점과 각 장단점은?

### 〈Database for Big Data〉

- NoSQL의 개념과 예제 시스템은?
- NOSQL database와 relational database의 주요 차이점은?
- Vector database가 AI 응용에 많이 사용되는 이유는? (2023년 가을 이후 일부 교육)
- LLM의 RAG에서 사용되는 벡터DB의 개념과 예제 시스템은?
- Vector database가 LLM의 RAG(retrieval-augmented generation)에 사용되는 원리는?

# 이산 수학, 확률, 통계

## <Keywords>

- Propositional Logic, Predicate Logic
- Program Verification
- Methods of Proof
- Sets
- Relations: Equivalence Relation, Order Relation
- Function
- Mathematical Induction
- Recursive Definition, Algorithms & Relations
- Counting
- Discrete Probability
- Graphs, Trees
- Algorithms

## <Program Verification>

- Loop Invariant란 무엇인가?
- Partial correctness와 Total correctness의 차이는 무엇인가?
- Precondition과 Postcondition은 무엇인가?

## <Relation>

- Relation의 정의는?
- $\forall a \in A, \forall b \in B$  일 때  $aRb \leftrightarrow (a, b) \in R$  이 뭔 뜻인지 설명하시오. 여기서 왼쪽 R와 오른쪽의 R이 같은 것인가, 다른 것인가? 다르다면 어떻게 다른가?
- Equivalence relation이란?
- Partial order relation이란? Total order relation이란?

## <Function>

- Bijection의 정의는?
- Bijective function은 반드시 total function인가?
- Partial function과 total function의 정의를 말하고 그 예를 드시오.

## <Mathematical Induction>

- Mathematical induction이란?
- Mathematical induction의 한 종류로서 Strong induction은 무엇인가?

### <Recursive Definition>

- Structural induction이란?
- Binary tree의 recursive 정의를 제시하시오.

### <Counting>

- $f: A \rightarrow B$  함수이며,  $|A|=n$  이고  $|B|=m$ 일때, 몇개의 onto function 을 정의 할 수 있는가?  
Inclusion-Exclusion 원리를 이용해 공역의 모든 원소가 정의역에 대응되는 경우의 수를 계산하면,

$$m^n - mC1(m-1)^n + mC2(m-2)^n - \dots = \sum_{k=0}^m mCk(-1)^k(m-k)^n$$

### <Discrete Probability>

- Random variable이 무엇인가?
- Sample space란 무엇인가?
- Sample space가 갖는 조건이 무엇인가?
- Exponential distribution이 무엇인가?
- Poisson distribution이 무엇인가?

### <Graph, Trees>

- Dijkstra의 Shortest path algorithm의 전제조건은?
- MST가 의미하는 것은?

# 네트워크

## <Keywords>

- Delay, Loss, Throughput
- HTTP, DNS, Video Stream
- UDP
- TCP, Congestion Control
- Network layer; IPv4, IPv6,
- Generalized forwarding, Routing
- Link Layer, wireless/mobile networks
- Security in computer networks

## <Preliminary>

- OSI 7계층에 대해 설명하시오.
- Transport, network, link 계층이 모두 연결에 관한 것인데, 어떤 특징과 차이가 있는가?

## <HTTP, DNS, Video Stream>

- HTTP/2에서 persistent connection이란?
- HTTP/3가 HTTP/2에 비해 바뀐 점은?
- DNS에서 주소에 대응되는 entry는 어떻게 찾아지는가?
- Video streaming에 쓰이는 프로토콜 중 하나를 설명하시오.

## <TCP>

- TCP와 같이 protocol을 reliable하게 설계 하려면 무엇이 필요한가?
- Flow control이란 무엇인가? Congestion control과 flow control의 차이점은?
- TCP와 UDP의 차이점은?
- Congestion이 발생했다는 것을 어떻게 알 수 있는가?
- 3-way handshaking이란?
- TCP congestion control은 어떤 원리로 동작하는가?
- TCP는 어떻게 발전하였는가?

## <Generalized forwarding, Routing>

- Generalized forwarding에 대해 설명하라.
- Internet routing은 어떻게 일어나는가?
- BGP가 어떻게 작동하는지 설명하여라.

### <Link Layer>

- CSMA/CD란?
- MAC address란? IP address가 MAC address에 비해 갖는 장점은?
- Wireless network에서는 왜 Collision avoidance를 채택하는가? 어떻게 하면 효율성을 증가시킬 수 있는가?
- 802.11 프로토콜의 특징을 설명하여라.

### <Security in Network>

- Network security의 4개 구성요소는 무엇인가?
- TLS에 대해 설명하여라.
- Network에서 checksum을 위해 사용하는 hash function과 security에서 data integrity를 위해 사용하는 hash function이 서로 interchangeable한가?

# Software Engineering

## <Keywords>

- Process Model: Heavy-weight, Prototyping, Agile, DevOps, ...
- Software Modeling: UML (Unified Modeling Language)
- Software Requirements Engineering
- Software Engineering Principles: Separation of Concerns, Abstraction, Modularity, ...
- Software Quality: Correctness, Robustness, Safety, Security, ...
- Software Design: Architectural Patterns, Resilient Design Patterns
- Software Testing: Black-box & white-box testing

## <Software Development Life Cycle and Process Model>

- 주어진 프로젝트에 적합한 프로세스모델을 선택하기위해 고려해야 할 사항은?
- Evolutionary Model (Prototyping)과 Incremental Model의 차이점에 대해 설명하고 어떠한 프로젝트에 적합한지?
- 기존의 Heavy process model의 단점을 해결하기위해 나온 Agile Methods들 중에 하나를 선택해 프로젝트에 적용할때 선제 조건은?

## <Software Modeling>

- UML 에서 시퀀스 다이어그램이란 무엇인가?
- 모델간의 추적성이란 무엇이고 추적성을 유지하는 것이 왜 중요한가?
- 추적성 유지와 관리를 위해 가장 중요한 모델은 무엇인가?
- UML에서 개발하고자하는 시스템의 동적, 정적, 인터랙션을 나타낼 수 있는 다이어그램에대해 예를 들어 설명하면?
- 유즈케이스 모델(Use Case Model)은 무엇을 표현하기 위한 것인가?
- 디자인의 평가 메트릭인 Cohesion 과 Coupling 에 대해 설명하시오.
- Software architecture란? 그 역할은?

## <Software Engineering Principles>

- SW에서 모듈화란? 모듈화를 잘 하기 위해 중요한 것은?
- DevOps는 어떠한 종류의 소프트웨어 개발과 배포에 효과적인가?
- AI 시대의 개발자의 가장 중요한 역할이 무엇이라고 생각하나?

## <Software Requirements Engineering>

- 요구 공학에서는 어떻게 요구사항을 수집하는가?
- 올바른 요구사항 파악을 위해 가장 중요한 것은?
- 요구사항 관리(Requirements Management)를 왜 해야 하는가?

- 품질속성(Quality Attributes)는 어떻게 도출하고 왜 중요한가?
- Requirement Engineering Process에서 Requirement Analysis의 역할에 대해 설명?
- Volatile requirement 원인과 해결방법은?

#### 〈Software Testing〉

- 테스트 기법으로 화이트 박스와 블랙 박스 테스트가 있는데, 그 둘의 차이점은? 화이트 박스 테스트가 에러가 없는 프로그램이라고 보장해 주지 않는 이유는?
- Verification Testing과 Validation Testing의 차이는 무엇인가?
- Test Oracle 문제란 무엇인가?
- Testing을 통해 결함이 없는 소프트웨어를 만들 수 있는가?

#### 〈기타〉

- CMMI 레벨이란?



# AI / ML

## <Keywords>

- Problem Solving: Searching, Constraint Satisfaction Problem
- First-Order Logic, Knowledge representation, Automated Planning
- Machine Learning: Decision Trees, Linear Regression, ...
- Deep Learning: FFN, CNN, RNN
- Reinforcement Learning
- Natural Language Processing, Computer Vision

## <Problem Solving>

- A\* 알고리즘에서는 goal까지의 거리를 예측하는 휴리스틱 함수가 쓰인다. 이 휴리스틱 함수는 어떤 조건을 만족해야 하는가?

## <Machine Learning>

- K-Means 목적함수와 알고리즘을 각각 설명하시오.
- Degree 4의 Polynomial Regression 모델이 오버피팅하고 있다면 모델의 Degree를 어떻게 바꾸어야 하는지, 왜 그렇게 하여야 하는지 설명하시오.
- 20세~40세 성인의 얼굴 사진을 보고 나이를 예측하는 모델을 만들려고 한다. Classification과 Regression 중 어떤 접근법이 적합한가? 왜 그런지 설명하시오.
- MNIST 데이터셋의 10-way classification 문제를 베이지안 방법론으로 모델링할 경우, 사전확률(Prior Distribution), 우도함수(Likelihood Function), 사후확률(Posterior Distribution)을 각각 어떻게 정의할지 설명하시오.
- Kernel Trick이란 무엇인지 간단히 설명하고, 왜 필요한지에 대해 설명하시오.
- Support Vector Machine(SVM)의 동작 원리에 대해 설명하시오 → 특정 데이터에서 학습된 SVM 모델이 있다고 했을 때, 이 데이터에서 support vector가 아닌 데이터를 모두 제거하고 학습한 SVM은 어떤 차이가 있을지 설명하시오.
- 확률변수의 조건부 독립이 무엇인지 설명하시오 → markov chain에 어떤 종류의 조건부 독립이 사용되는지 설명하시오 → markov chain의 조건부 독립이 문제가 되는 예시를 하나 이상 설명하시오

## <Deep Learning>

- 인공신경망을 학습할 때 사용하는 Dropout 테크닉이 무엇인지, 그리고 왜 필요한지 설명하시오.
- 트랜스포머(Transformer) 모델에서 Self-Attention 메커니즘이 무엇인지 설명하고, Query, Key, Value가 각각 어떤 역할을 하는지 설명하시오. 또한 Self-Attention이 RNN 기반 모델 대비 가지는 장점에 대해 설명하시오.
- 생성모델(Generative Model)이란 무엇인지 설명하고, 대표적인 생성모델 두 가지(예: VAE, GAN, Diffusion Model 등)를 선택하여 각각의 기본 아이디어와 학습 방식의 차이를 설명하시오.

- Convolution 연산은 무엇인지 설명하시오 → convolution layer가 fully-connected layer와 비교해 갖는 장점과 단점을 하나 이상 설명하시오.
- Inductive bias가 무엇인지 설명하시오 → CNN과 RNN에서 사용된 inductive bias가 무엇인지 각각 하나 이상 설명하시오.

### 〈Reinforcement Learning〉

- 강화학습에서 Exploration과 Exploitation의 개념이 무엇인지 각각 설명하시오.
- 강화학습에서 Value Function의 정의(수식)를 설명하고, State Value Function과 Action Value Function의 차이를 설명하시오. 또한 Value Function이 정책(Policy) 평가 및 개선에서 어떤 역할을 하는지 설명하시오.
- 강화학습에서 On-Policy와 Off-Policy 학습의 차이에 대해 설명하시오. 각각에서 사용되는 정책(behavior policy와 target policy)의 역할을 설명하고 대표적인 알고리즘을 예로 들어 두 방법의 차이를 설명하시오.

### 〈Graph Machine Learning and Mining〉

- 그래프 신경망(GNN)의 작동 원리에 대해 간단히 설명하고, 대표적인 모델을 하나 선정하여 동작 메커니즘을 간략히 설명하시오.
- 그래프 신경망이 Inductive Capability를 가질 수 있는 이유에 대해 간단히 설명하시오.
- 지식 그래프 임베딩이란 무엇이고, 왜 필요한지에 대해 간단히 설명하시오.
- 그래프 표현학습이란 무엇이고, 그 목적은 무엇인지 간단히 설명하시오.
- Graph Clustering의 대표적인 목적함수를 하나 설명하고, 해당 함수를 최적화하는 알고리즘에 대해 간략히 설명하시오.

### 〈Basic Math〉

- PCA를 통해 특정 저차원 공간으로 차원 축소를 한다는 것이 수학적으로 무엇을 의미하는지 설명하시오.
- Cross Entropy Loss를 설명하시오.
- 두 확률변수가 independent이면 covariance는 얼마인가요?
- Eigenvalue가 음수일 수 있나요?

# HCI & Human-AI Interaction

## <Keywords>

- Human Cognition: Mental Models, Attention, Perception, Memory
- Interaction: Affordance, Feedback, Mapping, Constraints, Interaction Design
- Usability & UX: Usability (Learnability, Efficiency, Safety), User Experience, Accessibility
- Design Methodology: User-Centered Design, Design Thinking, Prototyping, Iterative Design
- Evaluation: Usability Testing, A/B Testing, Surveys, Interviews, Experimentation
- Human-AI Interaction: Explainability, Interpretability, Trust, Transparency, Human-in-the-loop
- AI Issues: Bias, Fairness, Accountability, Overtrust, Undertrust
- Collaboration: Human-AI Collaboration, Mixed-Initiative Interaction, Personalization, Agency
- Affective & Social: Affective Computing, Emotion Recognition, Social Computing, Persuasive Technology
- Emerging Interaction: Ubiquitous Computing, Wearable Computing, Voice Interface, AR/VR

## <인터랙션>

- Affordance와 signifier의 차이는 무엇인가요?
- Mapping이 직관적이지 않을 때 사용자 경험은 어떻게 달라지나요?
- 좋은 feedback이란 무엇이며, feedback이 부족할 때 사용자는 어떤 문제를 겪는가?
- Affordance, signifier, feedback, mapping, constraint가 함께 작동하는 예를 하나 들어 설명하십시오.
- Error prevention과 error recovery의 차이는 무엇인가?
- 모델(Mental Model)의 정의가 무엇인가?
- 사용자의 멘탈 모델과 시스템 모델이 불일치할 때 발생하는 문제가 무엇인가?
- Voice interface의 장단점은 무엇인가? 어떤 상황에서 GUI vs. VUI가 좋은지?
- AR/VR이 기존 UI와 다른 점은 무엇인가?
- Context-aware interaction이란 무엇인가?
- Wearable computing에서 중요한 HCI 이슈는 무엇인가?
- AR/VR에서 presence, immersion, cybersickness는 무엇인가?
- AR/VR 인터페이스에서 spatial interaction이 중요한 이유는 무엇인가?

## <사용성/사용자 경험>

- Learnability, Efficiency, Safety 각각을 예시와 함께 설명하십시오.
- 사용성(Usability)을 정의하고, 주요 구성 요소는 무엇인가?

- 사용자의 첫 사용 경험과 장기 사용 경험은 어떻게 다르게 평가해야 하는가?
- 사용성과 사용자 경험(UX)의 차이가 무엇인가?
- 사용성이 좋은 시스템이 반드시 UX도 좋은가? 반례를 들어 설명하십시오.
- Universal design, inclusive design, accessibility의 차이를 설명하십시오.

#### <Design Methods>

- User-Centered Design의 핵심 원칙은 무엇인가?
- 왜 iterative design이 중요한가요?
- 프로토타입을 만드는 목적은 무엇인가?
- 프로토타입 평가 결과가 예상과 다를 때 어떻게 설계를 수정할 것인가?

#### <Evaluation>

- Think-aloud protocol은 무엇이며 장단점은 무엇인가?
- Usability testing과 A/B testing의 차이는 무엇인가?
- 인터뷰와 설문조사의 장단점을 비교
- 실험 설계에서 중요한 요소는 무엇인가?
- 실험 설계에서 독립변수, 종속변수, 통제변수는 무엇인가?
- HCI 실험에서 참가자 수, task design, counterbalancing이 중요한 이유는 무엇인가?

#### <인간-AI상호작용>

- HCI 관점에서 AI 시스템 사용 시 발생할 수 있는 주요 이슈는 무엇인가?
- Human-AI collaboration의 유형, 장점과 위험은 무엇인가?
- Mixed-initiative interaction이란 무엇인가? Mixed-initiative interaction에서 initiative를 언제 AI가 가져가고 언제 사용자가 가져가야 하는가?
- Human-in-the-loop가 필요한 상황과 필요하지 않은 상황을 구분하십시오.
- Human override 기능은 언제 필요하며, 어떤 한계가 있는가?
- AI 기반 추천 시스템에서 멘탈 모델 및 사용성/사용자 경험에 대해서 논의하십시오.
- AI가 불확실한 답을 낼 때 인터페이스는 이를 어떻게 표현해야 하는가?
- AI 시스템에서 사용자의 agency를 보장한다는 것은 무엇을 의미하는가? 사용자의 agency를 약화시키는 AI 인터페이스의 예는 무엇인가?
- AI bias는 어떻게 발생하며 어떻게 완화할 수 있는가? AI bias와 human bias는 어떻게 연결되는가?
- AI 시스템에서 fairness와 accuracy 사이의 trade-off를 어떻게 다룰 수 있는가?
- 사용자가 AI를 과도하게 신뢰하거나 불신하지 않도록 하기 위한 인터페이스 설계 방법은 무엇인가?
- AI 시스템에서 accountability는 어떻게 확보할 수 있나?
- Personalization이 사용자 경험에 미치는 영향은 무엇인가?
- Affective computing이란 무엇인가? Emotion recognition 기술의 한계는 무엇인가?

# 알고리즘

## <Keywords>

- (Time Complexity)
- Sorting
- Divide-and-conquer
- Dynamic Programming
- Greedy algorithms
- Graph algorithms
- Advanced data structures
- Complexity Theorem: NP-completeness

## <Sorting>

- Sorting algorithm 들에는 어떤 것들이 있는가? Insertion sort, heap sort, selection sort, quick sort, merge sort 중 optimal 한 것과 optimal 하지 않은 알고리즘은? 그 이유는?
- Partially ordered set이란?
- Quick sort 에 대해 설명하십시오.
- Show how you can make Quicksort to have  $O(n \lg n)$  worst-case running time.
- Optimal한 정렬 알고리즘의 예를 들어보세요. 그 알고리즘의 complexity를 분석하고 optimal하다는 것을 증명해보세요.
- $n$ 개의 숫자가 array로 주어졌습니다. 이 중에 Median을 찾는 알고리즘을 설명해보세요. 설명한 알고리즘의 complexity는 무엇입니까? Median을 찾는 optimal한 알고리즘의 complexity는 무엇입니까? optimal하다는 것을 어떻게 증명할 수 있나요? Median을 찾는 알고리즘을 이용하여  $n$ 개의 숫자를 정렬해보세요. 이 정렬 알고리즘 의 complexity는 무엇입니까? 두 가지 문제를 통해 Reduction이라는 개념을 설명해 보세요.

## <Dynamic Programming>

- Dynamic Programming이란?
- Dynamic programming과 Greedy algorithm에 대하여 각각이 무엇인지 설명하고, 공통점과 차이점을 이야기하세요. 구체적으로 Dynamic programming으로 풀 수 있는데 Greedy algorithm으로는 풀 수 없는 문제의 예를 들어보세요. 그 문제가 dynamic programming으로 풀 수 있다는 것을 증명해보세요.

## <Graph Algorithm>

- 그래프의 정의는? 트리의 정의는? 트리와 그래프의 관계는?
- Topological Sorting을 설명하십시오
- 오일러 사이클이란?

- Tree traversal의 3 가지 방식을 설명하시오.
- Transitive closure란?
- Finite state automata란?
- Binary tree란? binary tree에서 각 node는 두 개의 children을 갖거나 혹은 leaf node 이거나 둘 중에 하나라고 할 때, non-leaf node와 leaf node 개수의 관계식은 어떻게 되는가?
- Minimum spanning tree를 구하는 알고리즘과, 그 알고리즘의 복잡도는?
- T가 그래프 G의 minimum spanning tree라고 하자. G에 있는 임의의 두 정점 s와 t에 대해, G에서 s와 t 사이의 최단 경로가 반드시 T에 포함되는가?
- Dijkstra's algorithm을 설명하고 시간 복잡도를 분석하라.
- Floyd-Warshall algorithm을 설명하고 시간 복잡도를 분석하라.

#### 〈Advanced data structures〉

- Priority queue란? Heap이란?

#### 〈Complexity Theorem〉

- NP Complete의 정의는?

# 그래픽

## <Keywords>

- Geometric Objects and Transformation
- Rotation
- Interpolation
- Viewing
- Rasterization
- Ray Tracing

## <Transformation>

- Affine transform과 Rigid body transform에 대해서 차이를 설명하시오.
- Affine transform은 크게 두 부분으로 분리가 가능하다. 분리가 가능한 두 부분에 대해서 설명하시오.

## <Rotation>

- XYZ-Euler angle rotation의 Gimbal lock problem에 대해서 설명하시오.
- Rotation transform은 Quaternion multiplication으로 표현이 가능하다. Quaternion에 대해서 설명하시오.

## <Shading>

- 여러 shading 모델들과, 차이점을 설명하시오

## <Rendering>

- OpenGL rendering에서 model view matrix 에 대해서 설명하시오.
- OpenGL Shading Language는 크게 fragment shader와 vertex shader로 나누어 질 수 있다. 둘의 차이점에 대해서 설명하시오.
- 3차원 기하정보를 2차원 영상으로 표현하는 과정을 렌더링이라고 한다. 이를 위해서는 perspective projection이 필수적으로 요구되는데, 이에 대해 설명하시오.

## <Ray Tracing>

- 물체의 realistic 한 appearance를 렌더링하기 위해서는 물체의 표면 반사계수를 4차원 함수로 표현한 표면 모델(BRDF)이 필요하다. BRDF에 대해 설명하시오.
- Recursive ray tracing에 대해서 간단히 설명하시오.

## <기타>

- Edge detection에 대해 설명하시오.
- Massive한 데이터를 그래픽으로 출력하려면, 어떻게 해야 하는가?

- 3차원의 가상공간에 놓여 있는 3차원 물체들이 2차원의 그림으로 렌더링되어 우리의 화면에 보여진다. 이를 우리는 2차원 윈도우를 통해서 상호작용을 하게 되는데 이때 간단하게는 마우스를 이용하여 3차원 공간에 있는 물체를 선택하게 된다. 이때 마우스 input은 (x,y) 2차원 좌표인데 어떻게 3차원 공간에 있는 물체를 선택 (selection) 할 수 있을까? 이를 가능하게 할 수 있는 Object Selection 방법을 제시하라.
- 물체의 빠른 충돌 감지를 위해서 물체를 감싸는 간단한 도형인 bounding volume 을 hierarchical 하게 구성하여 사용한다. 이의 간단한 예를 만들어 얼마나 빠르게 처리를 할 수 있는지 설명하라.



# 정보 보호

## <Keywords>

- Confidentiality
- Integrity
- Public-key cryptosystem
- Blockchain
- Access Control
- Web Security
- Software Error & Vulnerabilities
- Program Analysis

## <Confidentiality>

- 암호 운영 모드에서 IV 또는 난수를 사용하는 이유를 설명하시오.

## <Integrity>

- 해시함수의 중요한 안전성 요구 조건을 세가지 설명하시오.
- 웹서버에서 사용자 패스워드를 안전하게 저장하고 사용하는 방법에 대해 설명하시오. 이에 필요한 암호학적 특성에 대해 논하시오.

## <Availability>

- SYN Cookie 에 대해서 설명하시오.
- DDoS 공격에 대응하기 위한 방어 기술에 대하여 설명하시오.

## <Public-key cryptosystem>

- 대칭키 암호와 공개키 암호의 장단점을 비교하시오.
- 이산대수(discrete log) 문제를 설명하시오.
- Diffie-Hellman 키교환 문제를 설명하고 어떻게 중간자공격(man-in-the-middle)에 취약한지 설명하시오.

## <Trust Model>

- 네트워크에서 정보를 전송할 때 데이터를 보호하기 위해 암호화하는 방법에는 어떤 것이 있는가?
- Transport Layer Security(TLS)가 보증하는 3가지 Security Properties에 대해서 설명하시오.
- PKI 시스템에서 CA(Certificate Authority)가 침해되었을때 발생할 수 있는 문제를 설명하시오.

### 〈Web security〉

- Cross-site Scripting 공격에 대해서 설명하고 이를 방어하기 위한 방법들에 대해서 설명하시오.
- Content-Security Policy를 이용하면 XSS 공격을 완벽히 방어하는것이 가능한가? 가능하지 않다면 그 이유에 대해서 설명하시오.

### 〈Software Error & Vulnerabilities〉

- Process 와 Thread에 대해서 설명하고 두개의 다른점을 설명하시오.
- Process 들이 이용하는 가상 메모리(Virtual Memory)가 서로 다른 Process간 공유된다면 장점과 단점은 무엇인가?
- Buffer overflow 공격에 대해서 설명하시오.
- Big endian and little endian에 대해서 설명하고 서로 다른 2개의 방식이 Buffer overflow 공격에 미치는 영향에 대해서 설명하시오.

### 〈Program Analysis〉

- 대표적인 정적, 동적 프로그램 테스트 기술에 대해 설명하고, 장단점을 논하시오.

### 〈기타〉

- 브라우저를 켜서 카이스트 웹사이트에 접속하기 까지 발생하는 모든 보안 기술들을 순서대로 논하고 간단하게 설명하시오.