

CSE Overview

A Basic CS Skill, ABC Winter School

박원

About Tutor

- 박원
- CSE 19
- 관심분야: Theoretic Computer Science, Combinatorics, Algorithmic Problem Solving
 - TCS \rightarrow Data Structures, Algorithm, etc.
 - Algorithmic Problem Solving \rightarrow BOJ
- 궁금한게 있다면 편하게 카톡/연락 주세요!

ABC 겨울학교 소개

Objective of 'A Basic CS Skills'

- 컴퓨터 공학과에서는 많은 툴을 다룰 수 있어야 함
 - git, vim, bash 등등
- 당장 2학년 1학기부터 사용
- But, 아무도 알려주지 않는다
- 그래서 미리 과에 적응하기 쉽게, CSE의 길잡이가 되는 프로그램

Plan of ABC Winter School

- 총 12회, 매주 2회
- 한번에 약 2~3시간으로 강의와 실습으로 구성
- 모든 만남은 **zoom**에서 이루어질 예정
- 모든 강의는 녹화 될 예정
- 최대한 출석해야하고, 사정이 있다면 미리 연락해주세요
- 수업 내용은 매주 [여기](#)의 [Readme.md](#)에 업데이트 예정
 - 모든 자료와 질의응답도 여기서 진행해요!

ABC Winter School Material

- [ABC Winter School - 2022 Winter, by Yunpyo, Ahn](#)
- [Our GitHub](#)

Introduction to CSE

What is CSE?

- **C**omputer **S**cience and **E**ngineering
- [전산학이란 무엇인가? by KAIST 강지훈 전산학부 조교수님](#)
- CSE만의 방법론으로 다양한 문제를 해결

Simple Problem

- 1부터 100사이의 수 N 이 있습니다.
- 이 N 이 얼마인지 맞추어야합니다.
- 여러분이 추측한 수 x 를 말하면, x 가 N 보다 큰지, 같은지, 작은지를 알려줍니다.
- 어떻게 하면 가장 빨리 맞출 수 있을까요?

Solution 1

- 큰지 작은지 답변은 무시하고, 1부터 100까지 전부 다 불러봅니다.
- 그럼 그중에 하나 아무튼 맞는게 있을거예요.
- 최악의 경우 100가지 숫자를 다 불러야합니다.

Solution 2

- 50을 부릅니다.
 - 50이 더 작다면 50과 100의 중앙에 있는 75를 부릅니다.
 - 그 반대라면 25를 부릅니다.
- 이를 계속 반복합니다...
- 기껏해야 7번 안에 N 이 얼마인지 맞출 수 있습니다.

Properties of CSE - Abstraction

- 현실의 문제를 추상화된 **훨씬** 표현이 단순하고, 일반적인 문제로 변형해서 생각한다
- 문제의 특성을 파악하기가 **훨씬** 쉬워진다!

Properties of CSE - Abstraction

예시

- 시간대별 주가의 목록이 주어져있을 때, 특정한 시간에 산 주식의 어느 정도 시간동안 떨어지지 않을까?
- [2, 3, 4, 1, 2] 라고 하면, 시간 1에 산 주식은 시간 4까지는 떨어지지 않는다.
- 배열의 어떤 위치 i 에 대해서, 그 위치 이후에 등장하면서, i 의 값보다 작은 가장 왼쪽에 있는 수를 찾는 문제로 추상화 할 수 있다.

Properties of CSE - Automation

- 추상화된 문제를 적절히 가공하면 우리가 원하는 정답을 얻을 수 있다.
- 컴퓨터가 알아서 자원, 서비스 등을 이해하고 처리

문제 해결의 과정

- 현실의 문제를 추상화 한다
- 추상 문제의 입력에 대해 자동화된 논리 구조에 따라, 추상화된 정답을 구한다.
- 그 정답을 현실의 정답으로 다시 바꾼다

Categories of CSE

- "by" computation
 - 세상에 존재 하는 문제를 컴퓨터과학적 방법론(computation)을 이 용해(by) 해결한다
- "of" computation
 - 컴퓨터과학적 방법론 그 자체에 대한 연구
- "for" computation
 - 컴퓨터과학적 방법론(computation)을 위한(for) 학문

UNIST CSE 소개

"by" computation

- 인간과 사회 문제를 어떻게 CS로 해결할까?
- 요즘 뜨겁디 뜨거운 Machine Learning
- 관련 교과목
 - 인건-컴퓨터 상호작용 개론, 정보시각화 기술
 - 인공지능, 기계학습, 컴퓨터 비전, 지능형로봇
 - 데이터베이스 시스템
 - 모바일 컴퓨팅
 - 컴퓨터 그래픽스

"of" computation

- 컴퓨터 공학 문제들의 추상적, 근본적 원리를 탐구
- 어떻게 하면 문제를 잘 해결할까?
 - 여기서 잘이란? → faster(시간복잡도), cheaper(공간복잡도)
- 관련 교과목
 - 고급프로그래밍
 - 데이터구조, 알고리즘, 계산이론
 - 프로그래밍 언어
 - 소프트웨어 공학

"for" computation

- 컴퓨터과학적 방법론은 당연히 컴퓨터 위에서 돌아갑니다.
- 어떻게 하면 더 효율적으로 컴퓨터 시스템을 구성할 수 있을까요?
- 관련 교과
 - 시스템프로그래밍, 컴퓨터 구조, 운영체제
 - 컴파일러개론, 병렬컴퓨팅
 - 컴퓨터 네트워크, 클라우드 컴퓨팅, 컴퓨터 보안

CSE Curriculum Map

5. Curriculum Map [교육과정 이수 체계도]

Sophomore		Junior		Senior	
Spring	Fall	Spring	Fall	Spring	Fall
Advanced Programming	Data Structures	Software Engineering	Artificial Intelligence	Parallel Computing	Introduction to Compilers
System Programming	Computer Architecture	Introduction to Algorithms	Computer Networks	Machine Learning	Mobile Computing
	Principles of Programming Languages	Operating Systems	Theory of Computation	Computer Graphics	Cloud Computing
		Introduction to Human Computer Interaction	Database Systems	Computer Security	Information Visualization
				Intelligent Robots	Computer Vision

※ The opening semesters for each courses can be changed depending on the department's circumstances.

Required Courses

Course Code	Course Title	Semester
CSE221	데이터구조	2
CSE241	고급 프로그래밍	1
CSE251	시스템 프로그래밍	1
CSE261	컴퓨터구조	2
CSE271	프로그래밍언어	2
CSE311	운영체제	1
CSE331	알고리즘	1
CSE351	컴퓨터네트워크	2
CSE401	졸업연구	-

전공 필수 과목 소개

CSE241 고급프로그래밍

- AIP에서 Python을 배웠다면, 이번에는 C++입니다.
- Python보다는 조금 더 어려운 내용과 함께, pointer, class 등의 내용을 포함하여, C++ 프로그래밍과 객체지향프로그래밍의 전반을 배웁니다.

CSE251 시스템 프로그래밍

- 컴퓨터는 전반적으로 어떻게 구성이 되어 실행되고 있을까요?
- 컴퓨터 시스템에 대한 전반적인 지식을 바로 이 과목에서 배웁니다.
- 이 과목은 추후에 들을 컴퓨터구조, 운영체제 및 기타 시스템 관련 전공선택 과목들의 지식의 기초가 되므로 잘 들어둘 수록 좋습니다!

CSE221 데이터구조

- 데이터를 어떻게 저장해두면 효과적으로 활용할 수 있을까요?
- 데이터를 저장하고, 가공하는 구조들을 배우는 수업입니다.
- 이후에 듣는 알고리즘 과목과 깊은 관련이 있고, 많은 과목들의 선이수 조건이 되는 과목입니다.

CSE261 컴퓨터 구조

- 시스템 프로그래밍에서 더 나아가 우리가 사용하고 있는 컴퓨터의 구조는 어떻게 되어있는지 배웁니다.
- 주로 간단한 형태의 cpu에서 연산들이 어떻게 실행되는지, 그동안 연 구자들이 속도를 높이려 어떤 노력을 해왔는지 배우게 됩니다.

CSE271 프로그래밍 언어

- 프로그래밍 언어에 대해서 배웁니다. 여러 프로그래밍 언어들의 특성을 배웁니다!
- 주로 배우는 것이 syntax와 semantic에 관한 것 입니다.
- 다양한 인터프리터와 분석기를 만들어봅니다.

CSE331 알고리즘

- 주어진 문제를 어떤 절차를 통해 해결할 수 있을지 배웁니다.
- 어떻게 하면 더 빠를까요? 어떻게 하면 더 메모리를 적게 사용할까요?
- 우리가 사용하는 알고리즘은 과연 올바른 알고리즘일까요?
- 어떻게 올바르게다는 것을 보일까요?

CSE311 운영체제

- 어떻게 하면 자원들을 효과적으로 활용할 수 있을까요?
- 이를 관리하는 운영체제에 대한 수업입니다.
- 유명한 과제인 Pintos를 배웁니다.

CSE351 컴퓨터 네트워크

- 여러대의 다양한 컴퓨터들 사이에서 통신은 어떻게 진행될까요?
- 어떻게 하면 보다 빠르고 많은 데이터를 전송할 수 있을까요?
- 어떻게하면 데이터의 손실 없이 데이터를 주고받을 수 있을까요?

전공 카탈로그

- 수업에서 이야기 하지 않은 더 자세한 졸업 요건은 아래에서 확인 가능합니다!
 - [2022 Undergraduate Course Catalog](#)
- 읽어보다가 질문이 있으면 자유롭게 물어봐주세요!

Questions?

Next:

What is WSL? How to install WSL?