Introduction to Algorithms

Course information

Instructor : Kilho Lee

Who Am I?

- Kilho Lee (이길호)
  - Venture hall #709
  - Homepage: https://sites.google.com/view/khlee
  - Email: khlee.cs@ssu.ac.kr
  - Office hours: by appointment

- **MISys lab**: Mobility Intelligence and Computing Systems Lab
  - Research interests
    - Autonomous vehicles
    - Cyber-physical systems/ Internet of Things
    - (Embedded) Real-time systems

Outline

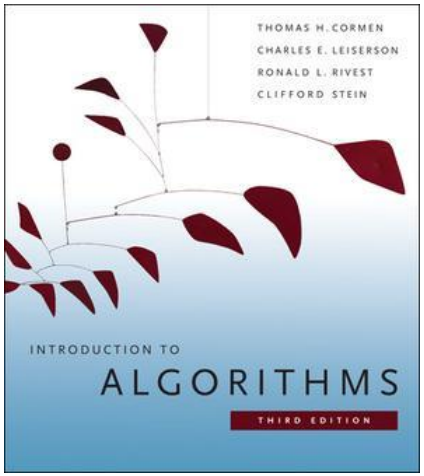
- Course Info
- Course overview

Course Information

- 알고리즘 (가)/(나), Introduction to Algorithms
  - 분반 통합 학점 부여
- Lecture time & place
  - Online, pre-recorded video clips.
  - 2~3 videos a week up to 150 min.
    - The LMS system (class.ssu.ac.kr)
- Prerequisites
  - Data Structures is strongly recommended
  - Programming languages:
    - **C++** for examples and programming assignments
    - Python for some examples instead of pseudo codes.

Course Information

- Textbook
  - Introduction to Algorithms (**CLRS**), Third Edition
    - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein
    - The latest edition is preferred, but any edition is fine
    - Available for free ebook via the SSU library
    - Translated version also available
  - 쉽게 배우는 알고리즘 (문병로)



- Acknowledgement
  - The lecture materials are built upon previous efforts of
    - Instructor David Eng (Stanford University)
    - Instructor Mary Wootters (Stanford University)
    - Professor H. Chwa (DGIST)

Course Information

- Grading
  - Homework: 20%
  - Midterm: 35%
  - Final: 35%
  - Class participation: 10%

Course Information

- Homework (20%)
  - 4~6 quizzes & programming assignments
    - PS (problem solving) style
  - Late submission penalties
    - Allowed to submit **up to 48 hours** after deadline.
    - Up to +24H (1d) : -25% penalty
    - Up to +48H (2d) : -50% penalty
    - Later than +48H: -100% penalty
  - We will be serious on any kind of plagiarism.

- Mid-term exam (35%), Final exam (35%)
  - Planned as **offline**, 대면 시험
  - Any kind of cheating will bring a serious penalty

Course Information

- Participation (10%)
  - Finish the video clip **at least 90%** within the period.
  - 각 컨텐츠의 학습 기간 이내에 청취 **(90% 이상)** 완료.
  - 결석 1회당 -1P
    - 지각 불인정. 실수/착오 (89.99% 청취, 0.1초 지각 등) 불인정.

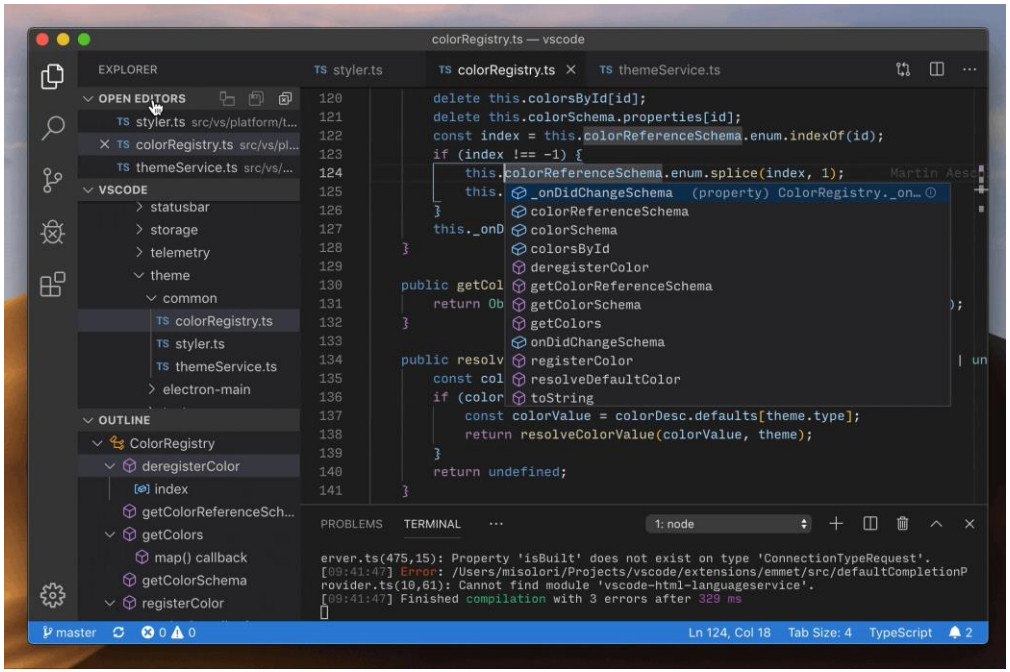
- Note
  - 교육부 지침에 따라 1/3 이상 결석 시 학점부여 불가 (F학점)
  - 유고 결석 시 SAINT 시스템을 통해 신청

Course Information

- Communication
  - By bulletin board (Q&A)
  - By email

Development environment setup

- VSCode + MinGW (g++)
  - 고급프로그래밍및실습 과목과 동일한 설정
  - Windows: LMS 영상 참고
  - MAC/Linux: VSCode/g++ 설정해서 사용 (LMS 영상 및 검색 활용) 하거나, 다른 개발환경 (vim/g++, sublime/g++ 등)



Introduction to Algorithms

L0. Overview

Instructor : Kilho Lee

Introduction to Algorithms

Computer Science

Computer Science is _____

13

Computer Science

Computer Science is abstraction

The single most important concept in computer science is abstraction

14

Abstraction

Abstraction is _____

15

Abstraction

Abstraction is to create a new model that allows to ignore irrelevant details

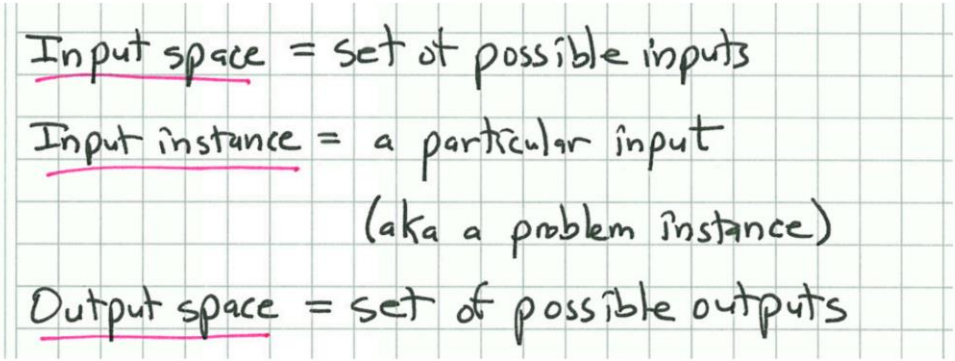
16

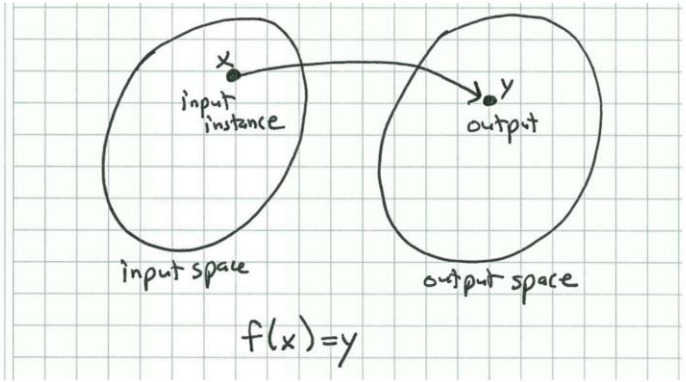
What is an Algorithm?

- Mathematical **abstraction** of a computer program
  - At the heart of programs lie algorithms
- Well-specified procedure for solving a computational problem

- Computational problem = mapping from inputs to outputs

Ex)
- Given array of integers, produce a sorted array
- Given a graph and nodes s and t, find a shortest path from s to t
- Given an integer, find its prime factors



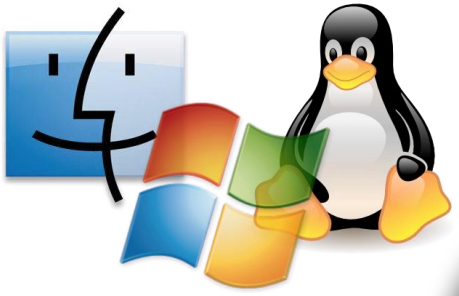


Why are you here?

- Algorithms are fundamental.
- Algorithms are useful.
- Algorithms are fun.
- It is a required course.

Why are you here?

- Algorithms are fundamental

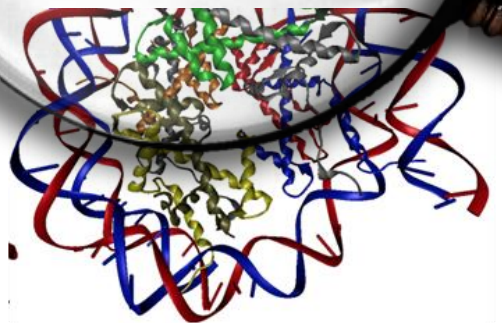


Operating Systems

Cryptography

The Computational Lens

Networking



Computational Biology

Mapping/Navigation
Google Search
Word Processing
Streaming Video
Games (graphics, rendering, ...)
Big data (querying, learning, ...)

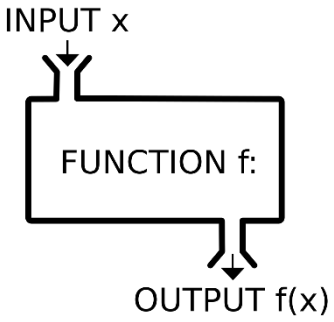
Why are you here?

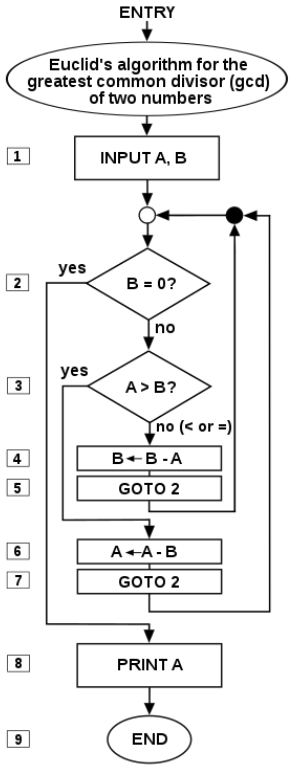
- Algorithms are useful
  - As we get more and more data and problem sizes get bigger and bigger, algorithms become more and more important.
  - Will help you get a job.

- Algorithms are fun
  - Algorithm design is both an art and a science.
    - Requires a combination of creativity and mathematical precision

Course Goals

- Formal definition of algorithm
  - An **algorithm** is a sequence of computational procedures that *transform the input into the output*
- *The **design** and **analysis** of algorithms*
  - These go hand-in-hand
- Implementation of algorithms





Course Goals

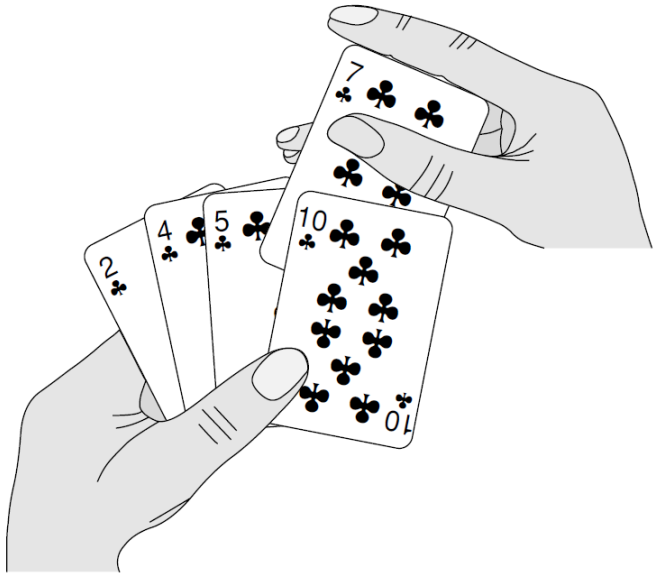
- **Analysis** of Algorithm
  - Does an algorithm actually work?
    - **Correctness**
  - Is it fast? (Does there exist any better algorithm?)
    - **Time/Space efficiency**
    - Lower bounds
    - Optimality

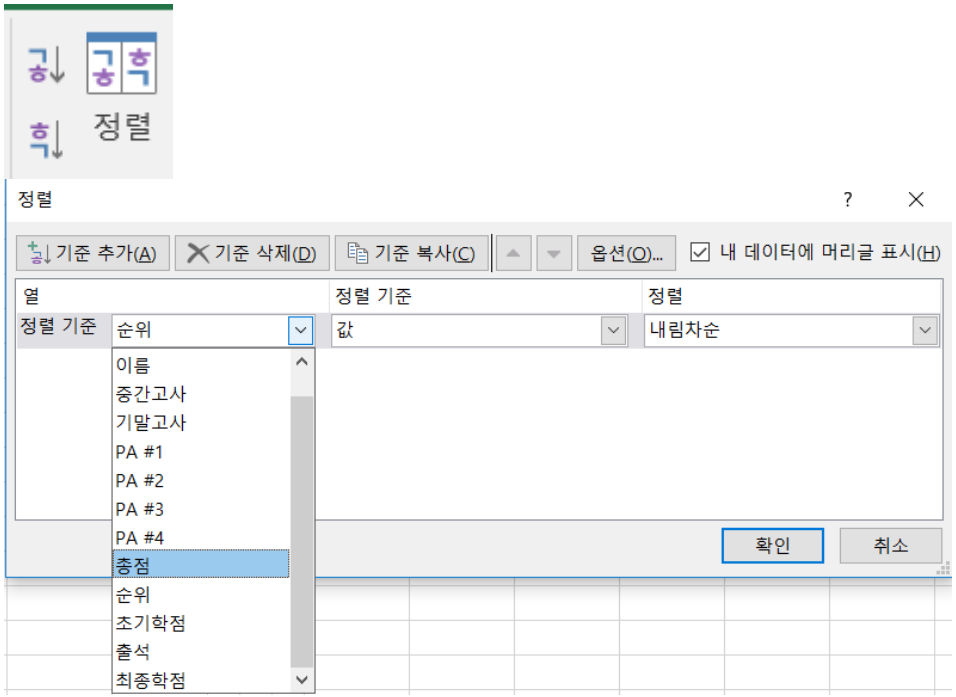
Course Topics

- Algorithmic Analysis
- Divide and Conquer
- Randomized Algorithms
- Tree Algorithms
- Graph Algorithms
- Dynamic Programming
- Greedy Algorithms
- Advanced Algorithms
- NP-completeness

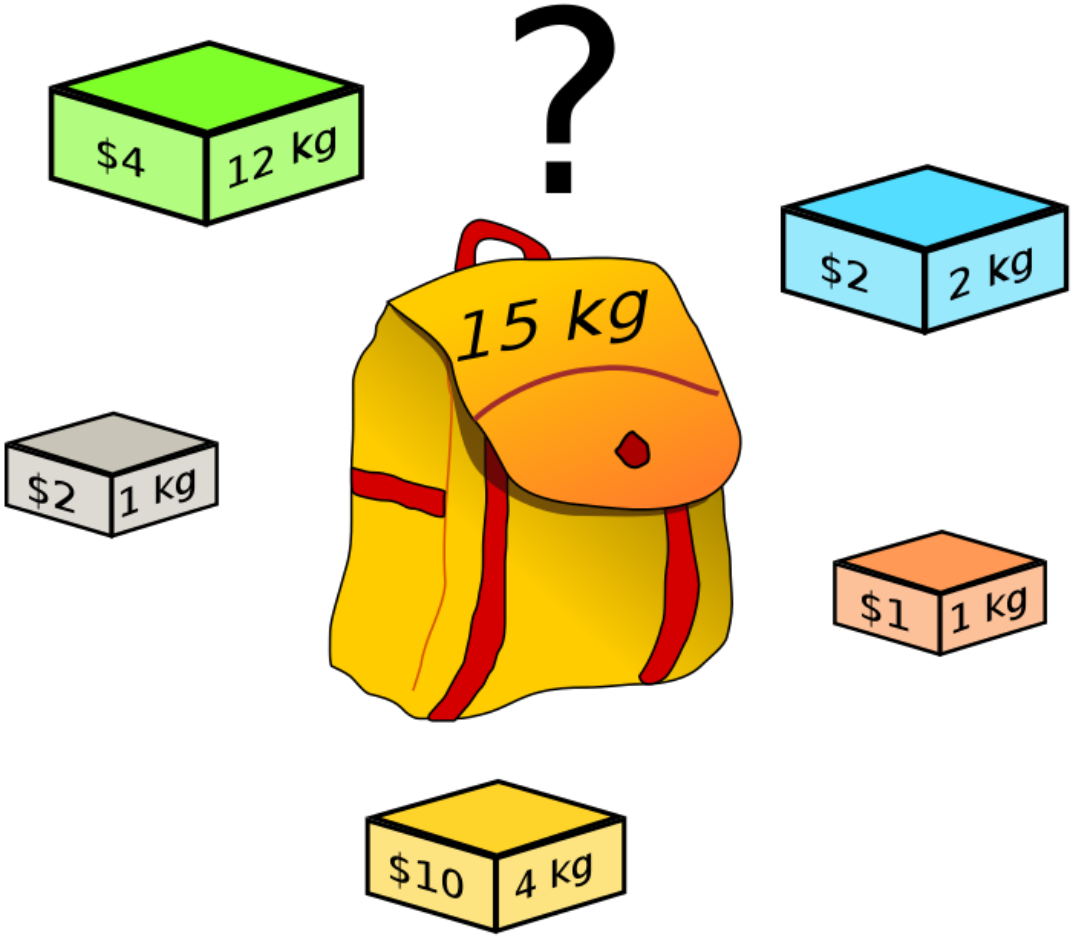
Sorting Problem

- Sorting problem
  - Input: A sequence of n numbers $< a_1, a_2, \ldots, a_n >$
  - Output: A permutation (reordering) of the input sequence, $< b_1, b_2, \ldots, b_n >$, such that $b_1 \leq b_2 \leq \cdots \leq b_n$
  - Instance: $< 7, 10, 4, 5, 2 >$





Knapsack Problem



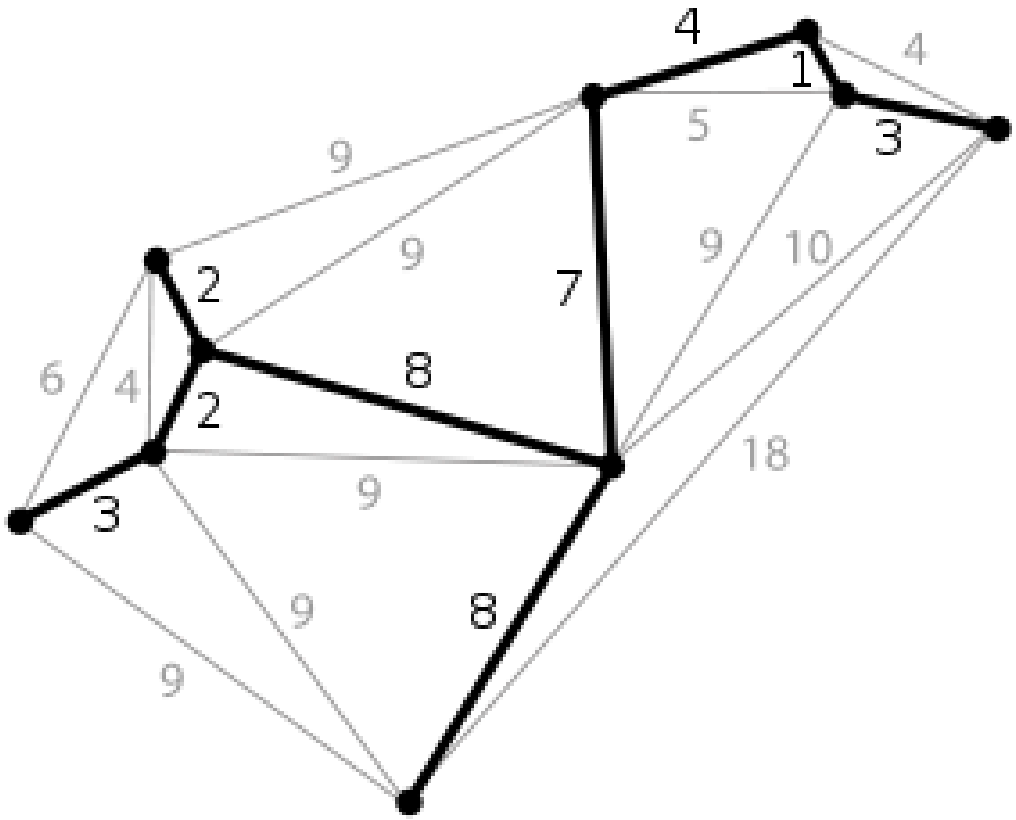
Figures from Wikipedia

Edit Distance

		m	o	n	k	e	y
	0	1	2	3	4	5	6
m	1	0	1	2	3	4	5
o	2	1	0	1	2	3	4
n	3	2	1	0	1	2	3
e	4	3	2	1	1	1	2
y	5	4	3	2	2	2	1

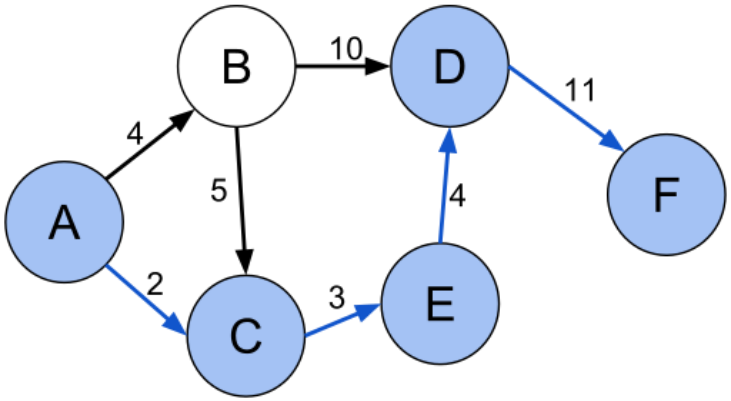
Figure from
https://vinayakgarg.wordpress.com/2012/12/10/edit-distance-using-dynamic-programming/

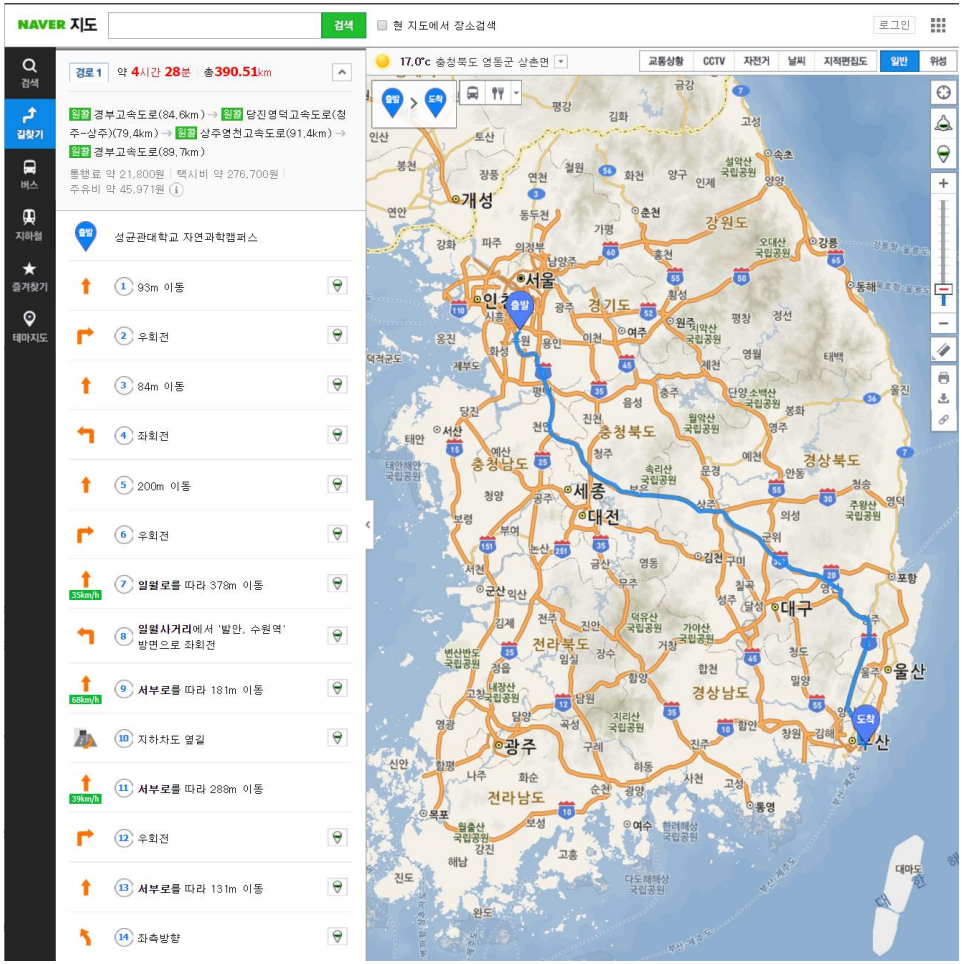
Minimum Spanning Trees (MST)



Figures from Wikipedia

Shortest Path

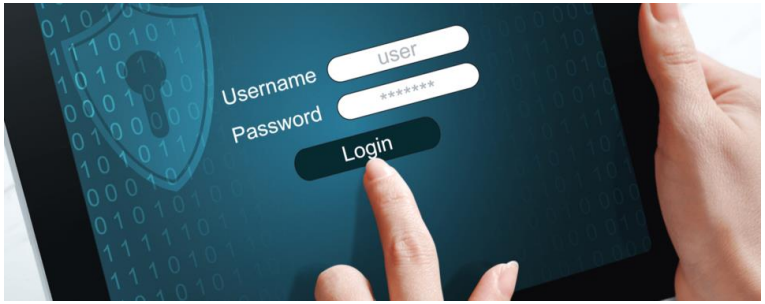


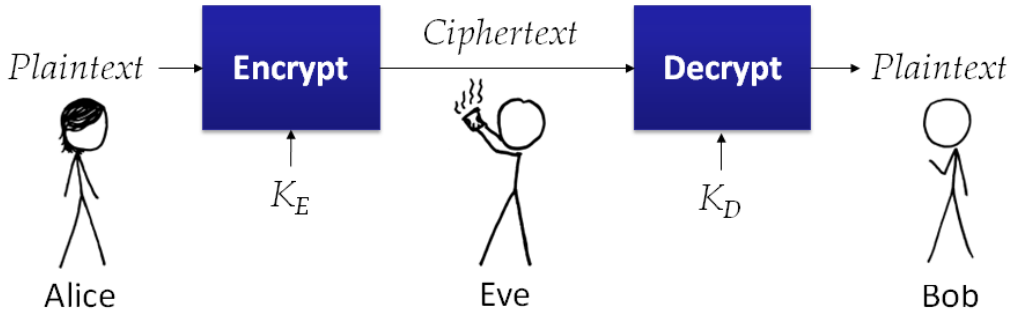


Figures from Wikipedia & NAVER

Security

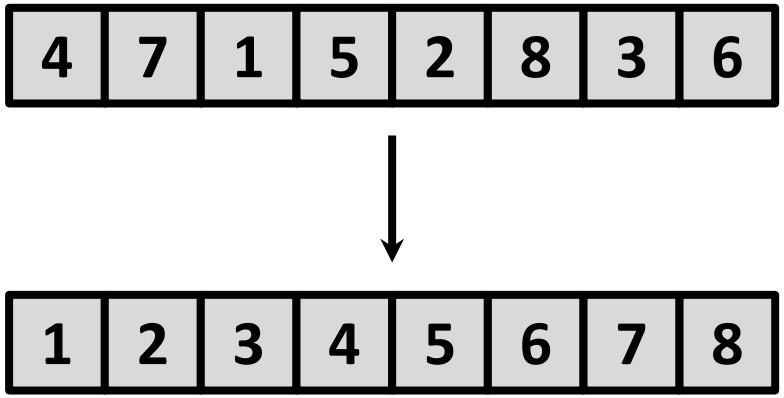
- Cryptography
- Authentication

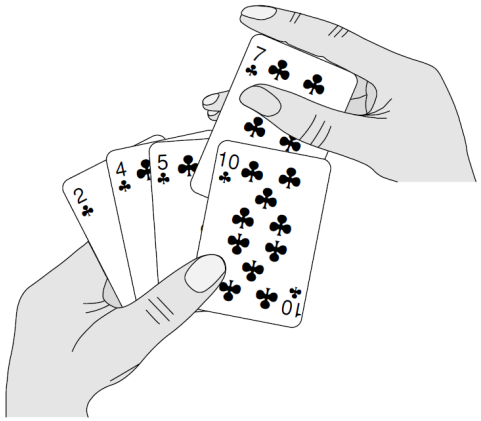




An example: Sorting

- Sorting algorithms order sequences of values.
  - For the sake of clarity, we'll pretend all elements are distinct.





A sorting algorithm

```
def sort(A):
  for i in range(1, len(A)):
    cur_value = A[i]
    j = i - 1
    while j >= 0 and A[j] > cur_value:
      A[j+1] = A[j]
      j -= 1
    A[j+1] = cur_value
```

Summary

- Course Goals
  - What is an algorithm?
  - Why we study algorithms?
    - Fundamental, useful, fun
  - Goal: the design and analysis of algorithms