# Course Overview

Data Structures

C++ for C Coders

한동대학교 김영섭 교수

idebtor@gmail.com

# Data Structures & C++ for C Coders

- *Overview*
    - *Why study data structures?*
- *Syllabus*
    - *Piazza*
    - *GitHub*
    - *Atom*
    - *MinGW/MSYS*
    - g++ (GNU GCC C++ Compiler)
    - MS Visual Studio

# Course overview

**What does the data structure mean?**

- **Data structures**:
  - **methods to store and organize data** in a computer so that it can be used efficiently.
  - A key to designing efficient **algorithms**.

- **Algorithms**:
  - methods for solving a problem

- **Data structures &algorithms** are the fundamentals of programming.
  - To become a good computer scientist or engineering it is essential to master the **data structures and algorithms** and learn to apply them to the real world problems.

which is complicated or complex.

# Course overview

**What is this course?**

- Intermediate-level course.
- Programming **after** programming for problem solving with applications.

| topic | data structures and algorithms |
|---|---|
| concepts | algorithms, time-complexity, array and structure |
| **data types** | linked list, array, stack, queue, trees,<br>union-find, bag, priority queues |
| sorting | selection sort, quick sort, merge sort, heap sort |
| searching | binary search tree, hashing |
| graph | BFS, DFS |

# Why study data structures?

**Their impact is broad and far-reaching**

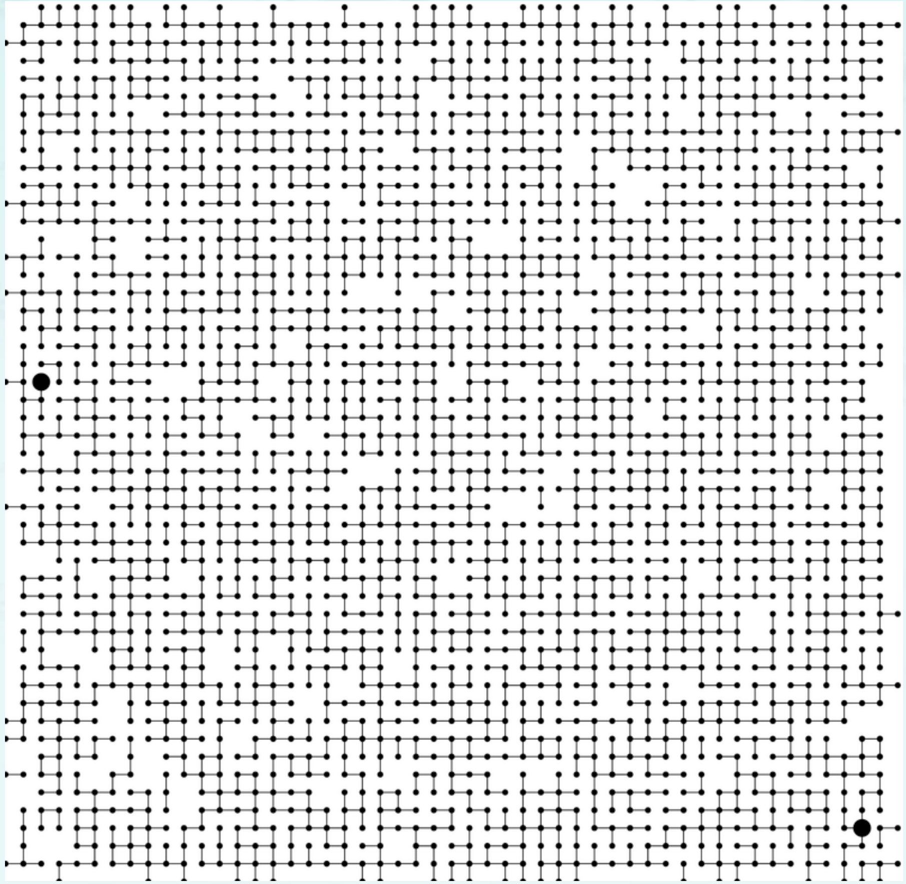
- **Internet** Web search, packet routing, distributed file sharing, ...
- **Social networks** News feeds, advertisements, ...
- **Computers** Circuit layout, file system, compilers, ...
- **Computer graphics** Movies, video games, virtual reality, ...
- **Multimedia** MP3, JPG, DivX, HDTV, face recognition, ...

- **Security** Cell phones, e-commerce, voting machines, ...
- **Biology** Human genome project, protein folding, ...
- **Physics** N-body simulation, particle collision simulation, ...

# Why study data structures?

**To solve problems that could not otherwise be addressed**

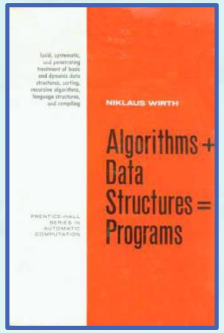
- To work with **algorithms** to solve problems
- Ex. Network connectivity, navigation



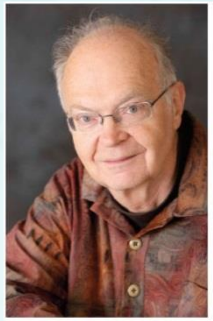
# Why study data structures?

**To become a proficient programmer.**

*" Algorithms + **Data Structures** = Programs. " —* Niklaus Wirth



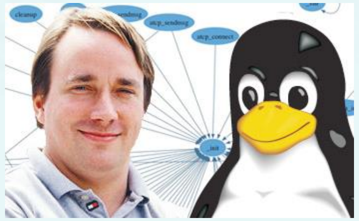
*" An **algorithm** must be seen to be believed. " —* Donald Knuth



Donald E. Knuth, winner of the Katayanagi Prize for Research Excellence.

*" I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about **data structures** and their **relationships**. "*

— Linus Torvalds *(creator of Linux)*



# Why study data structures?

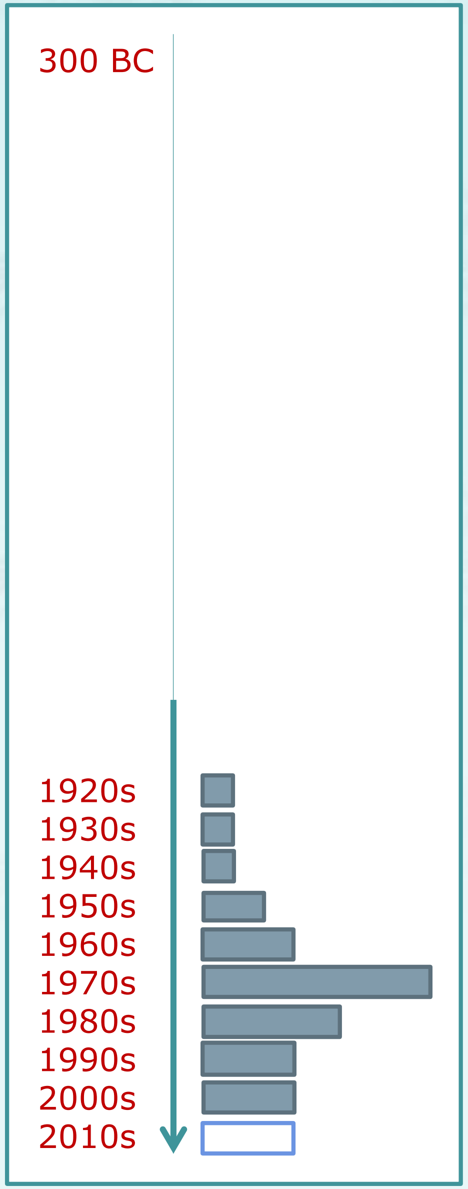
**Algorithms – Old roots, new opportunities.**

- Study of **algorithms** dates at least to Euclid.
- Formalized by Church and Turing in 1930s.
- Some important **algorithms** were discovered by undergraduates in a course like this.
- Then, why **data structures**?
  It always comes with algorithms like its shadow.

Ex. Fast Fourier Transform(FFT) Algorithm

Joseph Fourier(1768-1830) used for heat-transfer computation.
1805 – invented by Carl Friedrich Gauss.
1965 – popularized by James Cooley(IBM) and John Tukey(Princeton).

1986 – JPEG(Joint Photographic Experts Group) was formed.
1992 – issued the first standard of JPEG using DCT
Discrete cosine transform – another form of FFT.



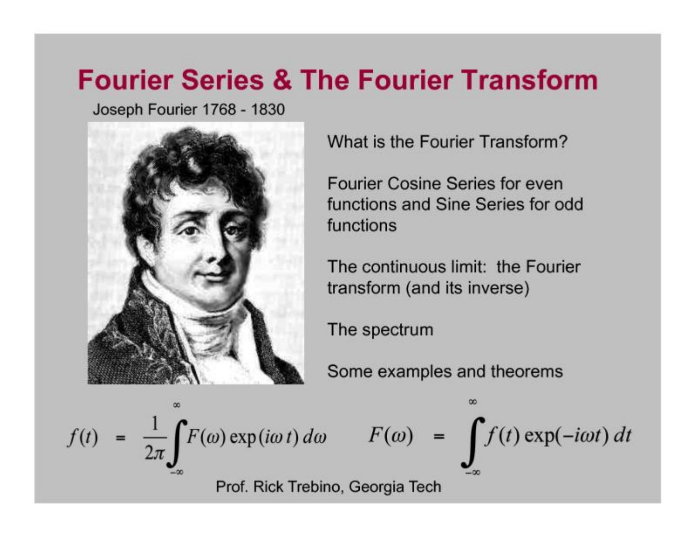
# Why study data structures?

**They may unlock the secrets of life and of the universe.**

Computational models are replacing math models in scientific inquiry.

Ex. Fourier Transform → Fast FT algorithm → Image Processing → JPEG/MPEG

1805 1965 1992



## Fourier Series & The Fourier Transform

Joseph Fourier 1768 - 1830

What is the Fourier Transform?

Fourier Cosine Series for even functions and Sine Series for odd functions

The continuous limit: the Fourier transform (and its inverse)

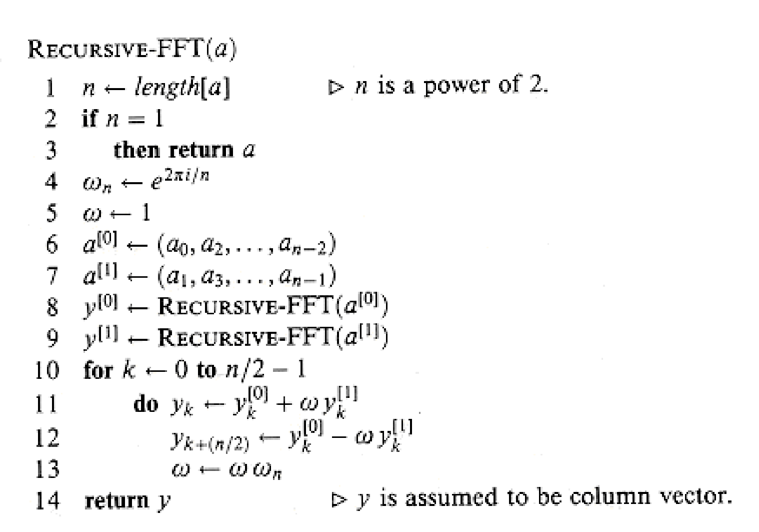
The spectrum

Some examples and theorems

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\exp(i\omega t)\,d\omega \qquad F(\omega) = \int_{-\infty}^{\infty} f(t)\exp(-i\omega t)\,dt$$

Prof. Rick Trebino, Georgia Tech

~ old century science
(formula based)



```
RECURSIVE-FFT($a$)
$n \leftarrow length[a]$                 ▷ $n$ is a power of 2.
if $n = 1$
    then return $a$
$\omega_n \leftarrow e^{2\pi i/n}$
$\omega \leftarrow 1$
$a^{[0]} \leftarrow (a_0, a_2, \ldots, a_{n-2})$
$a^{[1]} \leftarrow (a_1, a_3, \ldots, a_{n-1})$
$y^{[0]} \leftarrow$ RECURSIVE-FFT($a^{[0]}$)
$y^{[1]} \leftarrow$ RECURSIVE-FFT($a^{[1]}$)
for $k \leftarrow 0$ to $n/2 - 1$
    do $y_k \leftarrow y_k^{[0]} + \omega\, y_k^{[1]}$
       $y_{k+(n/2)} \leftarrow y_k^{[0]} - \omega\, y_k^{[1]}$
       $\omega \leftarrow \omega\, \omega_n$
return $y$                          ▷ $y$ is assumed to be column vector.
```

21th century science
(algorithm based)

# Why study data structures?

- Their impact is broad and far-reaching.
- Old roots, new opportunities.
- To solve problems that could not otherwise be addressed.
- For intellectual stimulation.
- To become a proficient programmer.
- They may unlock the secrets of life and of the universe.
- For fun and profit..

***Data Structures!***
*Why study anything else?*

*Algorithm!*
*Why study anything else?*

# Why study data structures?

## Textbook & resources – required

- Fundamentals of data structures, 2nd Edition
  by Horwitz, Sahni, Anderson

## Prerequisites

- C Programming: loops, arrays, functions, recursion, **pointer**
- C programming: using it extensively and **seriously.**
- Mathematics: high-school algebra.

## Programming environment

- Piazza & GitHub
- MinGW/MSYS & GNU GCC C++ compiler
- Atom
- Later.... MS Visual Studio Community

## Required reading

- Course Syllabus, GitHub/idebtor/nowic/GettingStarted

# ITP20001/ECE 20010 Data Structures

## Data Structures

### Chapter 1

- *overview*
  - *pointers and dynamic memory allocation*
- *algorithm specification*
  - *recursive algorithm*
- *data abstraction*
- ***performance analysis - time complexity***
  - ***discrete math***

*Youngsup Kim, idebtor@handong.edu, Handong Global University*