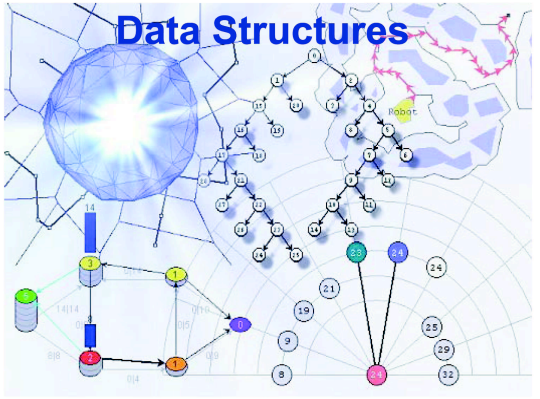
# BBM 201
# DATA STRUCTURES

## Lecture 1:
## Basic concepts for data structures




2016-2017 Fall

# About the course

- This course will help students understand the **basic data structures** such as matrices, stacks, queues, linked lists, etc.

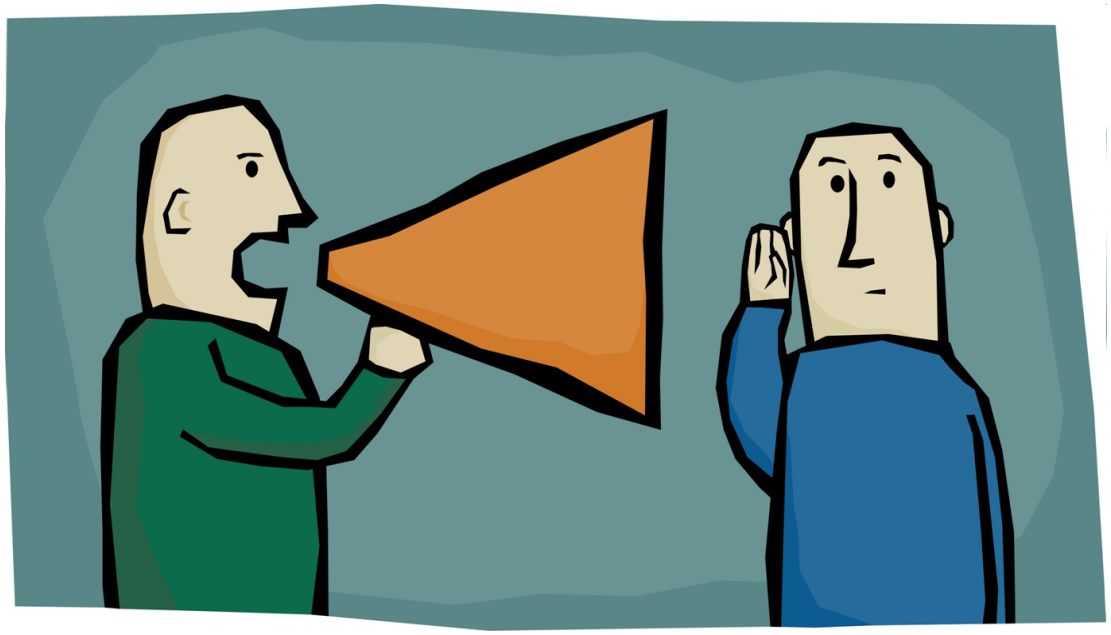
- **BBM 203 Programming Laboratory:** The students will gain hand-on experience via a set of programming assignments supplied as complementary.

- **Requirements**: You must know basic programming (i.e. BBM101).

# References


- Data Structures Notes, Mustafa Ege.
- Fundamentals of Data Structures in C. Ellis Horowitz and Sartaj Sahni, 1993.
- Data Structures A Pseudocode Approach with C. Richard F. Gilberg, Behrouz A. Forouzan
- Data Structures and Algorithm Analysis in C++. Mark Allen Weiss.
- Problem Solving and Program Design in C, 7th Edition. Jeri Hanly and Elliot Koffman, Pearson, 2013
- The C Programming Language, 2nd Edition. Brian Kernighan and Dennis Ritchie, Prentice Hall, 1988
- Practical C Programming 3rd Edition. Steve Oualline, O'Reilly Media, 1997

# Communication



- The course web page will be updated regularly throughout the semester with lecture notes, programming assignments, announcements and important deadlines.

http://web.cs.hacettepe.edu.tr/~bbm201

# Getting Help

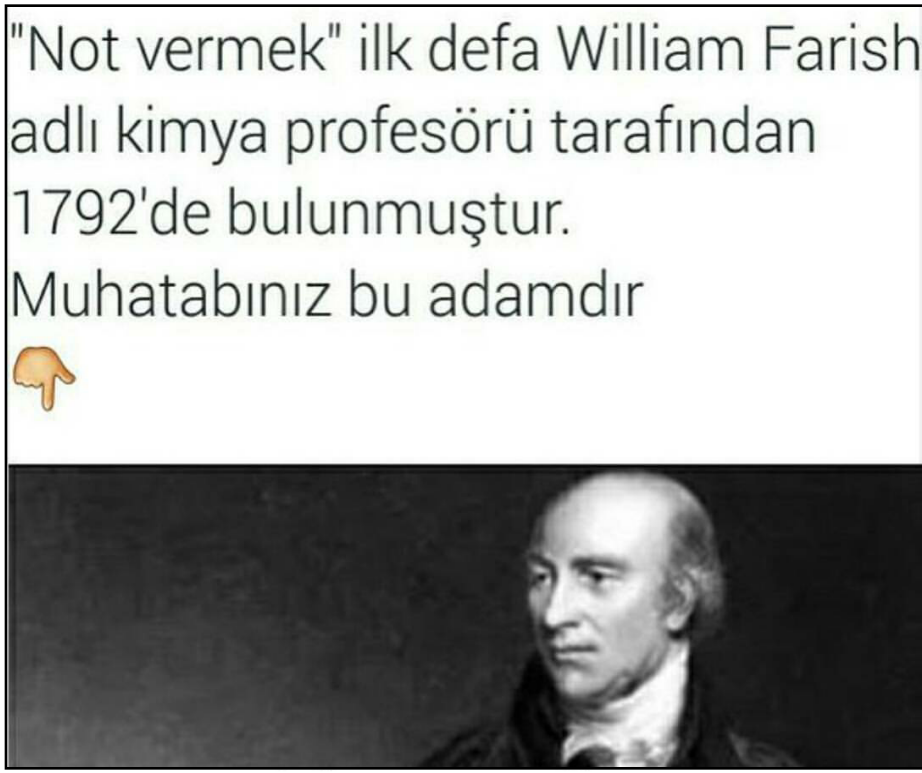
- **Office hours**
  - See the web page for details
- **BBM 203 Programming Laboratory**
  - Course related recitations, practice with example codes, etc.
- **Communication**
  - Announcements and course related discussions through piazza

BBM 201: https://piazza.com/hacettepe.edu.tr/fall2016/bbm201

BBM 203: https://piazza.com/hacettepe.edu.tr/fall2016/bbm203

# Course Work and Grading

- **2 midterm exams (50%)**
  - Closed book and notes
- **Final exam (45%)**
  - Closed book
  - To be scheduled by the registrar
- **Class Attendance (5%)**
  - Attempting to create false attendance (e.g., signing in the attendance list on behalf of someone else) will be punished.
  - Attendance is mandatory – students who fail to attend more than %30 of the lectures will fail from the course (≈if you do not attend 4 lectures, you will fail).



# Course Overview

| Week | Date | Title |
|---|---|---|
| 1 | 7-Oct | Introduction |
| 2 | 14-Oct | Recursion, Performance analysis |
| 3 | 21-Oct | Arrays and Matrices |
| 4 | 28-Oct | Stacks&Queues |
| 5 | 4-Nov | Midterm Exam 1 |
| 6 | 11-Nov | Evaluation of expressions |
| 7 | 18-Nov | Linked Lists |
| 8 | 25-Nov | Doubly Linked Lists |
| 9 | 2-Dec | Sparse Matrix using Circular LL |
| 10 | 9-Dec | Midterm Exam 2 |
| 11 | 16-Dec | Trees |
| 12 | 23-Dec | Binary Search Trees |
| 13 | 30-Dec | Graphs |
| 14 | 6-Jan | Review |

# BBM 203 Programming Laboratory I

- **Programming assignments (PAs)**
  - Four assignments throughout the semester.
  - Each assignment has a well-defined goal such as solving a specific problem.
  - You **must work alone** on all assignments stated unless otherwise.

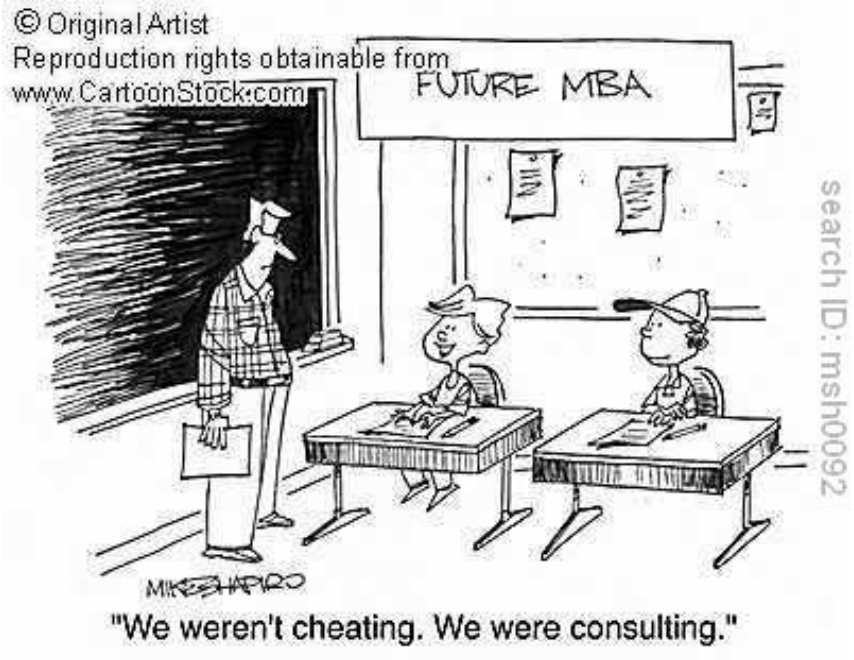
- **Important Dates**
  - See the course web page for schedule.

# Policies

- **Work groups**
  - You must work alone on all assignments stated unless otherwise
- **Submission**
  - Assignments due at 23:59 (no extensions!)
  - Electronic submissions (no exceptions!)
- **Lateness penalties**
  - No late submission is accepted

# Cheating

- **What is cheating?**
  - Sharing code: by copying, retyping, looking at, or supplying a file
  - Coaching: helping your friend to write a programming assignment, line by line
  - Copying code from previous course or from elsewhere on WWW




- **What is NOT cheating?**
  - Explaining how to use systems or tools
  - Helping others with high-level design issues

# Cheating

- **Penalty for cheating:**
  - Suspension from school for 6 months (minimum)

YÜKSEKÖĞRETİM KURULU

- **Detection of cheating:**
  - We do check: Our tools for doing this are much better than most cheaters think!

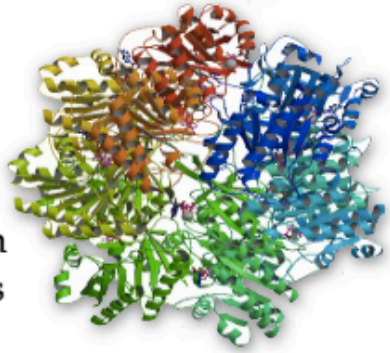
# BASIC CONCEPTS
# FOR DATA STRUCTURES

# Digital Data


Movies


Music

Photos



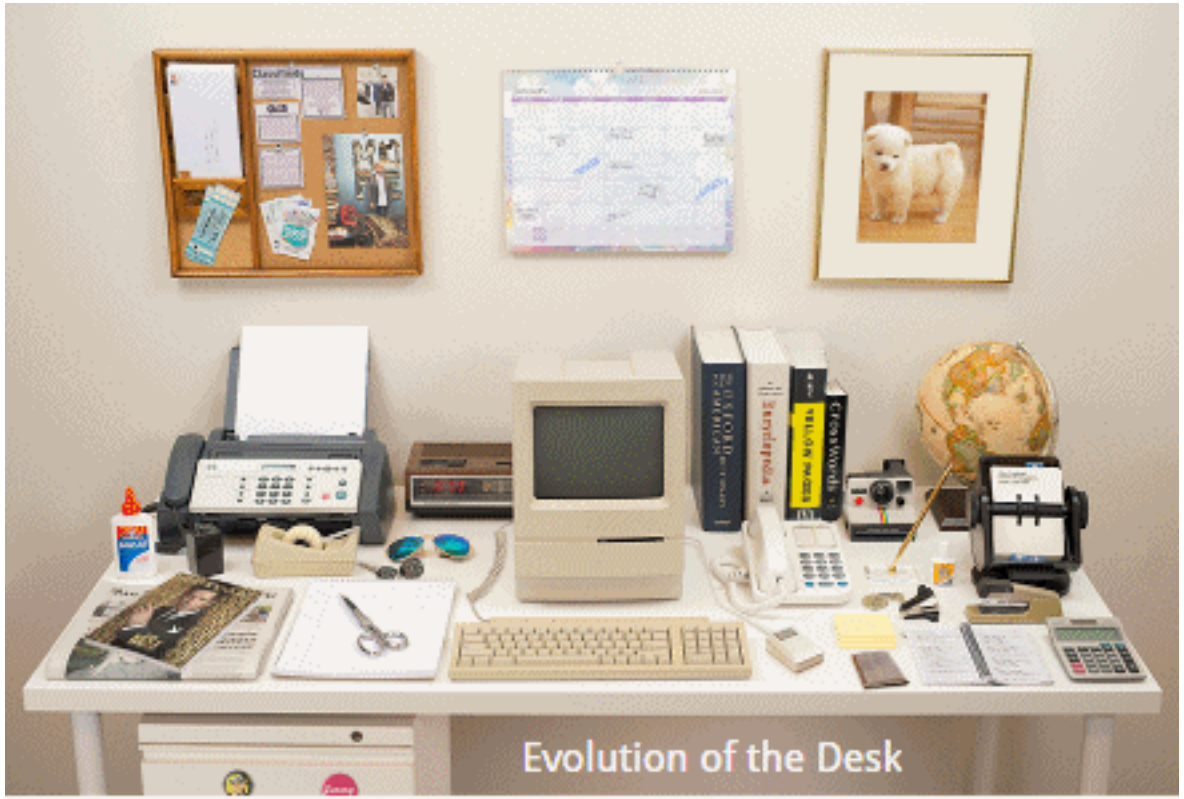
Protein Shapes

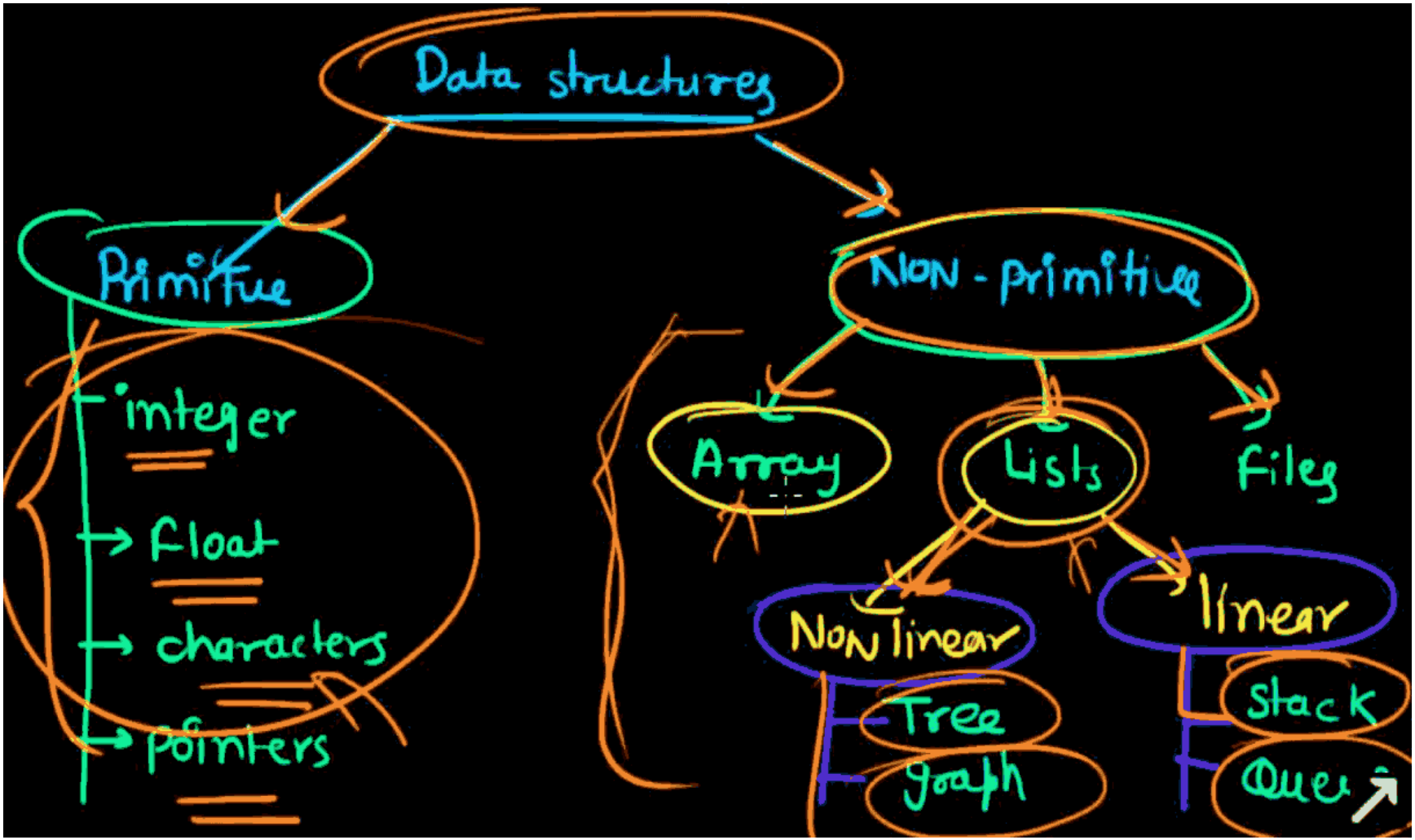

Maps

DNA

```
gatcttttta tttaaacgat ctctttatta gatctcttat taggatcatg atcctctgtg
gataagtgat tattcacatg gcagatcata taattaagga ggatcgtttg ttgtgagtga
ccggtgatcg tattgcgtat aagctgggat ctaaatggca tgttatgcac agtcactcgg
cagaatcaag gttgttatgt ggatatctac tggttttacc ctgcttttaa gcatagttat
acacattcgt tcgcgcgatc tttgagctaa ttagagtaaa ttaatccaat ctttgaccca
```

00|0|0|00|0|0|0|0|0|00|00|00|0|0|00000|00|00|0|00....


Evolution of the Desk
1980


Data structures
Primitive
integer
float
characters
pointers
Non-primitive
Array
Lists
Files
Non linear
Tree
Graph
linear
Stack
Queue


1
2
3
4
5
6
7

# Digital Data Must Be ...

- Encoded (e.g. 01001001 <->  )

- Arranged
  - Stored in an orderly way in memory / disk

- Accessed
  - Insert new data
  - Remove old data
  - Find data matching some condition

} The focus of this class

- Processed
  - Algorithms: shortest path, minimum cut, FFT, ...

## Data Structures→Data StructurING

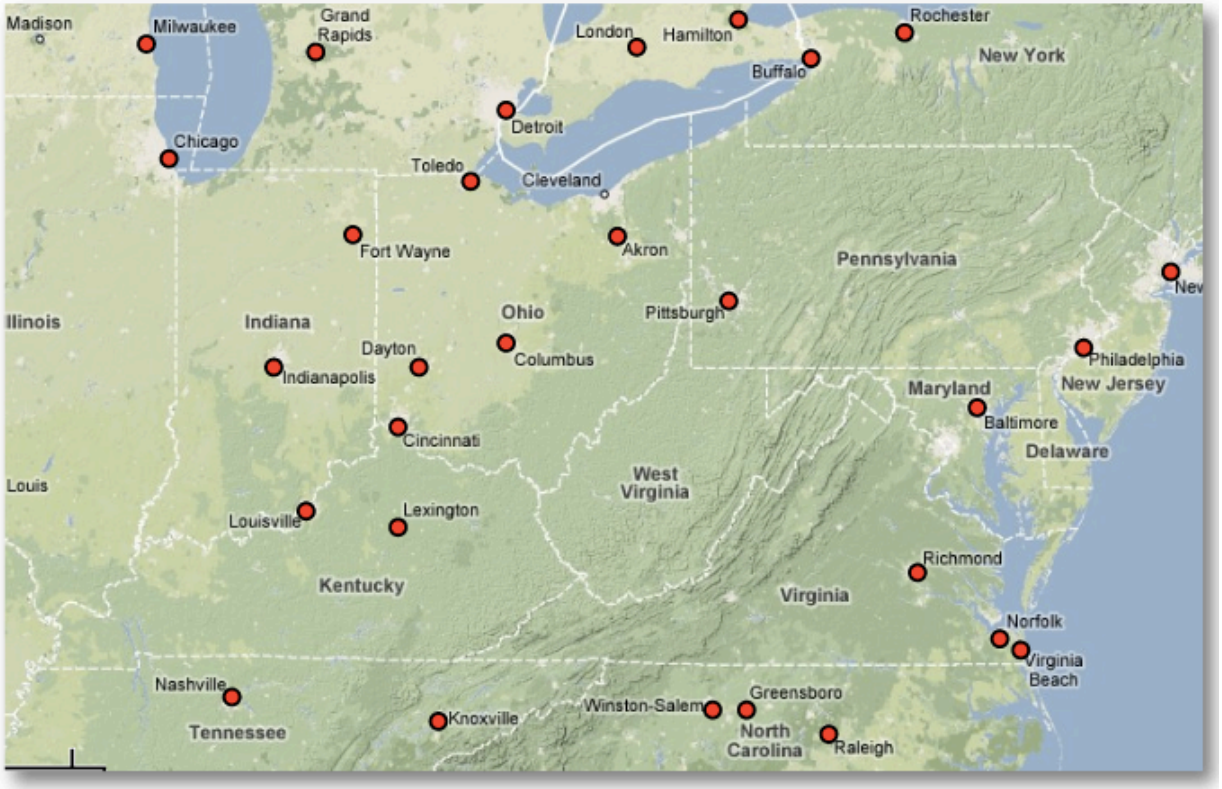
How do we organize information so that we can find, update, add, and delete portions of it **efficiently**?

# Data Structure Example Applications

- How does Google quickly find web pages that contain a search term?
- What's the fastest way to broadcast a message to a network of computers?
- How can a subsequence of DNA be quickly found within the genome?
- How does your operating system track which memory (disk or RAM) is free?
- In the game Half-Life, how can the computer determine which parts of the scene are visible?

# Suppose You're Google Maps…

- You want to store data about cities (location, elevation, population)…




What kind of operations should your data structure(s) support?

# Operations to support the following scenario…

Finding addresses on map?

- *Lookup city by name...*

Mobile user?

- *Find nearest point to me...*

Car GPS system?

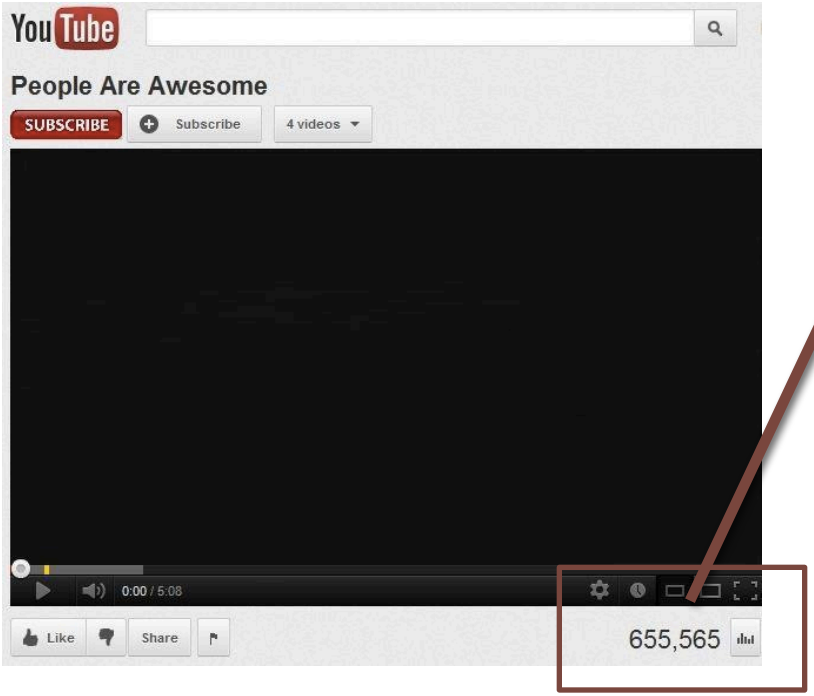
- *Calculate shortest-path between cities…*
- *Show cities within a given window…*

Political revolution?

- *Insert, delete, rename cities*

# How will you count user views on YouTube?

- Lets write a userViewCount() function

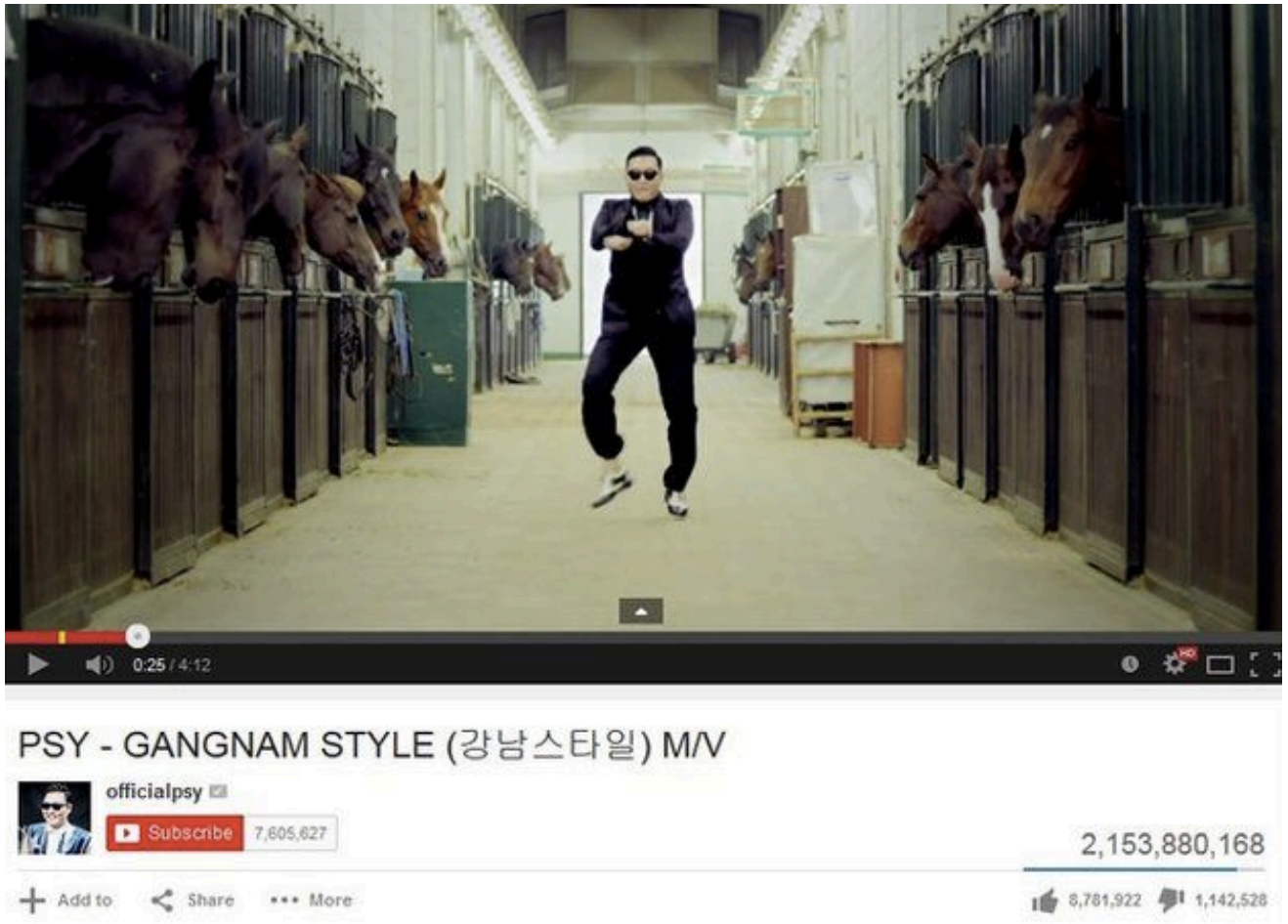



```
int userViewCount (int
current_count)
{
        int new_count;
        new_count =current_count + 1;
        return new_count;
}
```

Will this implementation work all the time?

# How will you count user views on YouTube?

%99.9 times yes.



# How will you count user views on YouTube?

BBC Sign in | News | Sport | Weather | Shop | Earth | Travel

NEWS

Home | Video | World | UK | Business | Tech | Science | Magazine | Entertainment & Arts

Asia | China | India

Asia

## Gangnam Style music video 'broke' YouTube view limit

4 December 2014 | Asia

http://www.bbc.com/news/world-asia-30288542

The Economist | World politics | Business & finance | Economics | Science & technology | Culture

**The Economist explains**

Explaining the world, daily

Previous | Next | Latest The Economist explains | All latest updates

The Economist explains

## How "Gangnam Style" broke YouTube's counter

Dec 10th 2014, 23:50 BY G.F. | SEATTLE

Timekeeper | Like 6.1k | Tweet 114

http://www.economist.com/blogs/economist-explains/2014/12/economist-explains-6

**YouTube's counter previously used a 32-bit integer**

YouTube said the video - its most watched ever - has been viewed more than 2,147,483,647 times.

It has now changed the maximum view limit to 9,223,372,036,854,775,808, or more than nine quintillion.

# How bad can it be?

- June 4, 1996
- Ariane 5 rocket launched by the European Space Agency
- After a decade of development costing $7 Billion

(~21 Billion in Turkish Liras, just for comparison Istanbul's third bridge cost estimates are 4.5 Billion TL)

- Exploded just 40 seconds after its lift-off
- The destroyed rocket and its cargo were valued at $500 million
- Reason?

# How bad can it be?

- Reason?
- Inertial reference system error: specifically a 64 bit floating point number relating to the horizontal velocity of the rocket with respect to the platform was converted to a 16 bit signed integer.
- The number was larger than 32,767, the largest integer storable in a 16 bit signed integer, and thus the conversion failed.
- $500 Million rocket/cargo
- Time and effort

# Goals

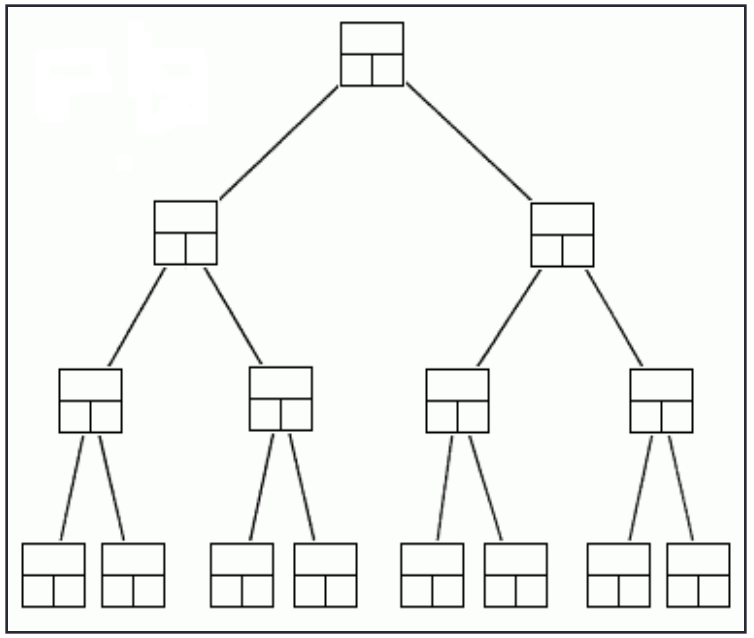
"I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships."

Linus Torvalds, 2006

# Data Structures

A data structure is a way to store and organize data in computer, so that it can be used *efficiently*.

Some of the more commonly used data structures include lists, *arrays, stacks, queues, heaps, trees, and graphs*.



Binary Tree

# What are data structures?

- Data structures are software artifacts that allow data to be stored, organized and accessed.

- Ultimately data structures have two core functions: put stuff in and take stuff out.

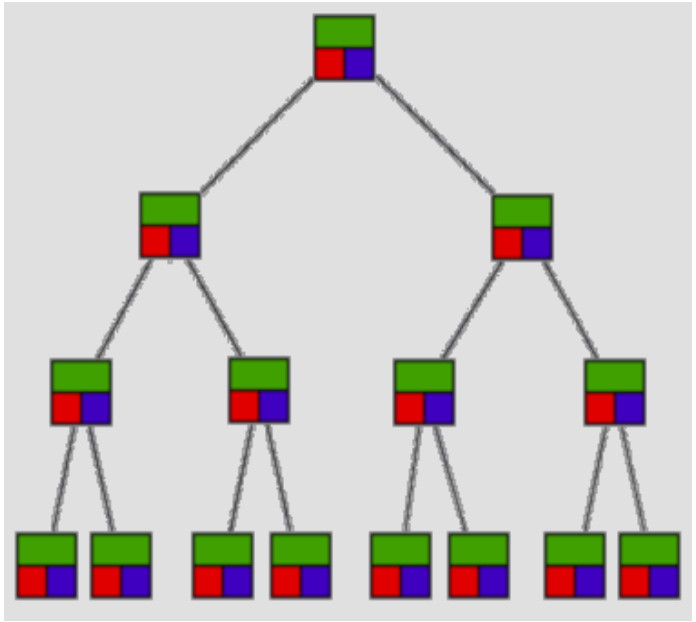
# Why so many?

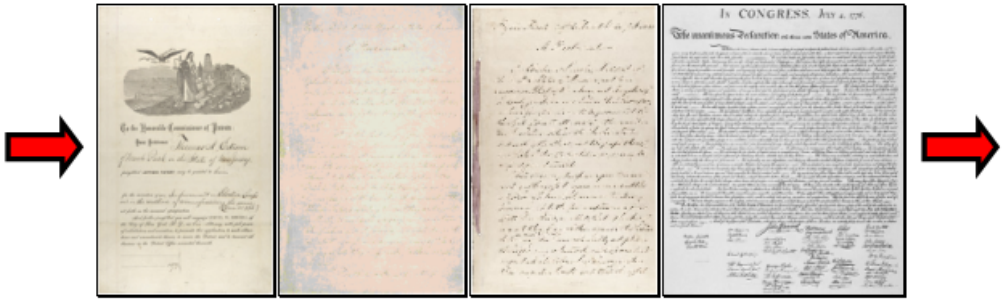
- Space efficiency
- Time efficiency:
  - Store
  - Search
  - Retrieve
  - Remove
  - Clone etc.

# Choosing Data Structures

Queue vs Binary Tree

---Which one to use for what task?





# Why So Many Data Structures?

- Ideal data structure:
  - “fast”, “elegant”, memory efficient
- Generates tensions:
  - time *vs.* space
  - performance *vs.* elegance
  - generality *vs.* simplicity
  - one operation’s performance *vs.* another’s

*The study of data structures is the study of tradeoffs. That’s why we have so many of them!*