

# BBM 201

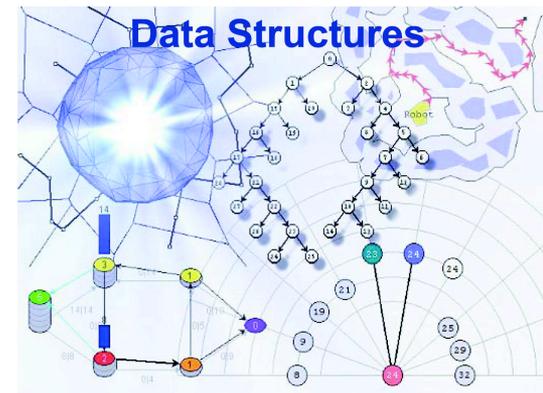
# DATA STRUCTURES

---

Lecture 1:  
Basic concepts for data structures



2017-2018 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 and Algorithm Analysis in C++. Mark Allen Weiss.
- 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
- 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 
- BBM 201: <https://piazza.com/hacettepe.edu.tr/fall2017/bbm201>
- BBM 203: <https://piazza.com/hacettepe.edu.tr/fall2017/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).



The joy of learning

# Course Overview

| Week | Date    | Topic  |
|------|---------|--|
| 1    | Sept 28 | <a href="#"><u>Introduction to Data Structures</u></a>                       |
| 2    | Oct 5   | <a href="#"><u>Recursion and Performance Analysis</u></a>                    |
| 3    | Oct 12  | <a href="#"><u>Arrays</u></a> & <a href="#"><u>Matrices</u></a>              |
| 4    | Oct 19  | <a href="#"><u>Arrays</u></a> & <a href="#"><u>Matrices (Cont.)</u></a>      |
| 5    | Oct 26  | <a href="#"><u>Stacks and Queues</u></a>                                     |
| 6    | Nov 2   | Midterm Exam 1   |
| 7    | Nov 9   | <a href="#"><u>Evaluation of Expressions</u></a>                             |
| 8    | Nov 16  | <a href="#"><u>Linked Lists 1</u></a> & <a href="#"><u>2</u></a>             |
| 9    | Nov 23  | <a href="#"><u>Linked List Applications (Stack, Queue, Hashtable)</u></a>    |
| 10   | Nov 30  | <a href="#"><u>Doubly Linked Lists</u></a> & <a href="#"><u>Examples</u></a> |
| 11   | Dec 7   | Trees  |
| 12   | Dec 14  | Midterm Exam 2   |
| 13   | Dec 21  | <a href="#"><u>Trie &amp; Graphs</u></a>                                     |
| 14   | Dec 28  | <a href="#"><u>Graphs</u></a>  |

# BBM 203 Programming Laboratory I

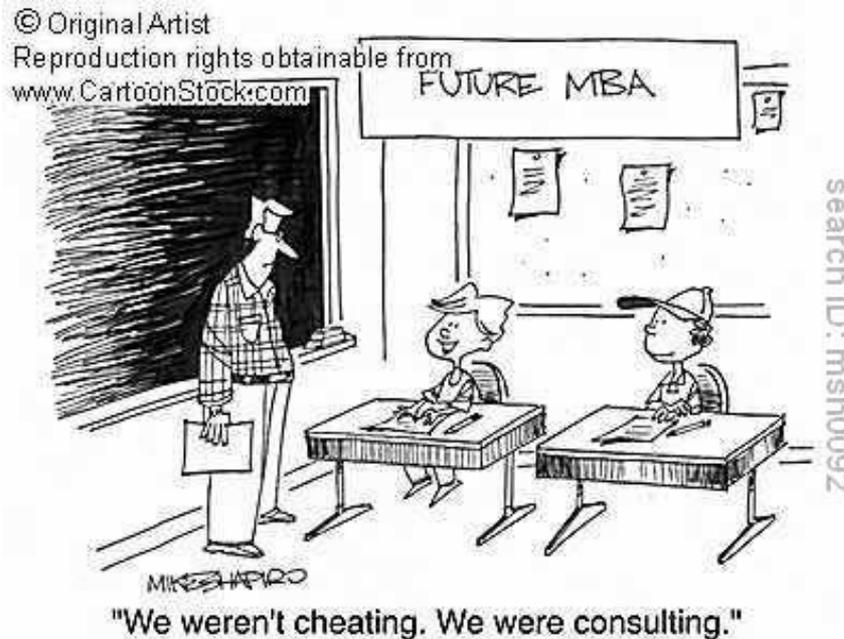
- **Programming assignments (PAs)**
  - Four assignments throughout the semester.
  - Each assignment has a well-defined goal such as solving a specific problem.
  - You **must work alone** on all assignments stated unless otherwise.
- **Important Dates**
  - Programming Assignment 1 9 October 2017
  - Programming Assignment 2 23 October 2017
  - Programming Assignment 3 13 November 2017
  - Programming Assignment 4 4 December 2017

# 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)



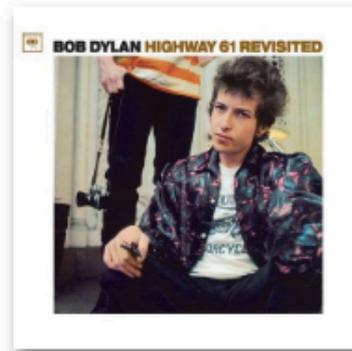
- **Detection of cheating:**
  - ◆ We do check: Our tools for doing this are much better than most cheaters think!

# **BASIC CONCEPTS FOR DATA STRUCTURES**

# Digital Data



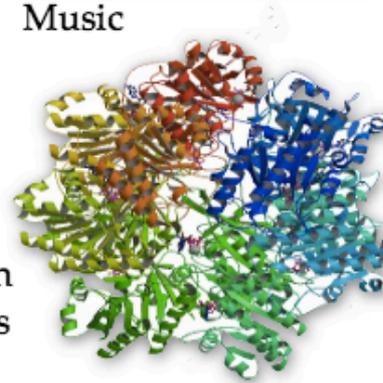
Movies



Music



Photos



Protein Shapes

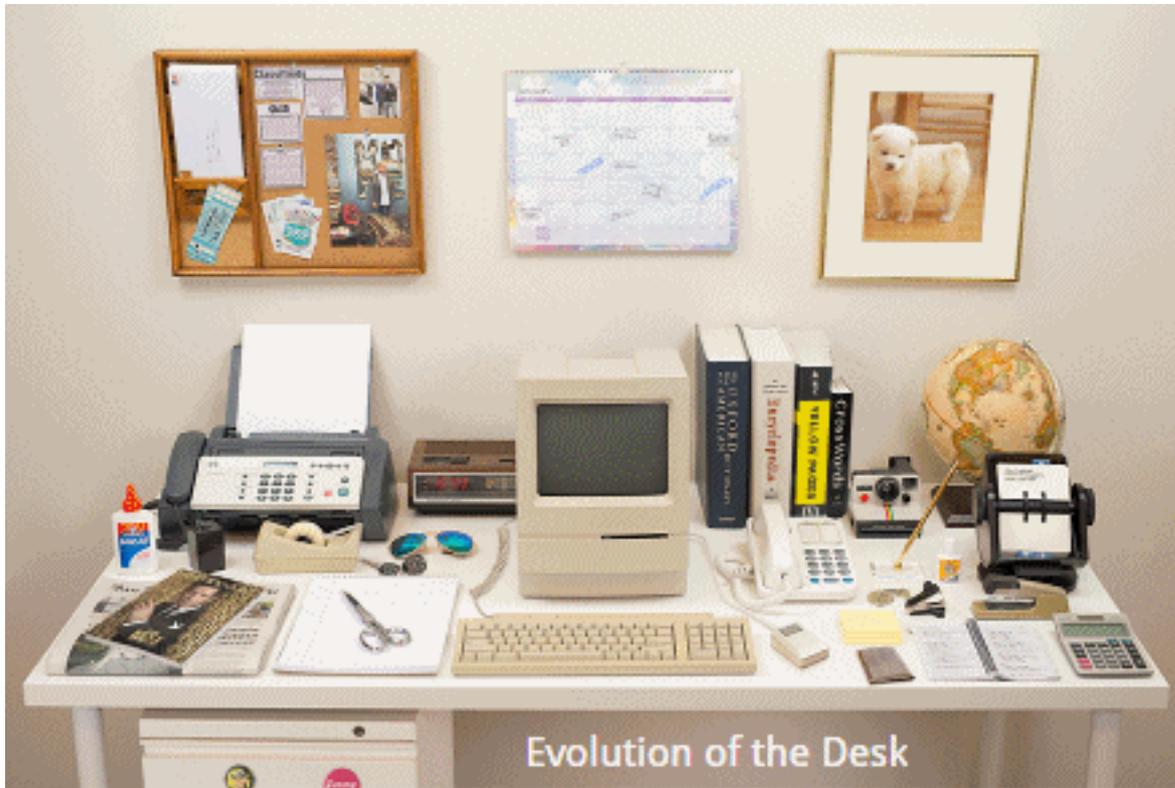


Maps

## DNA

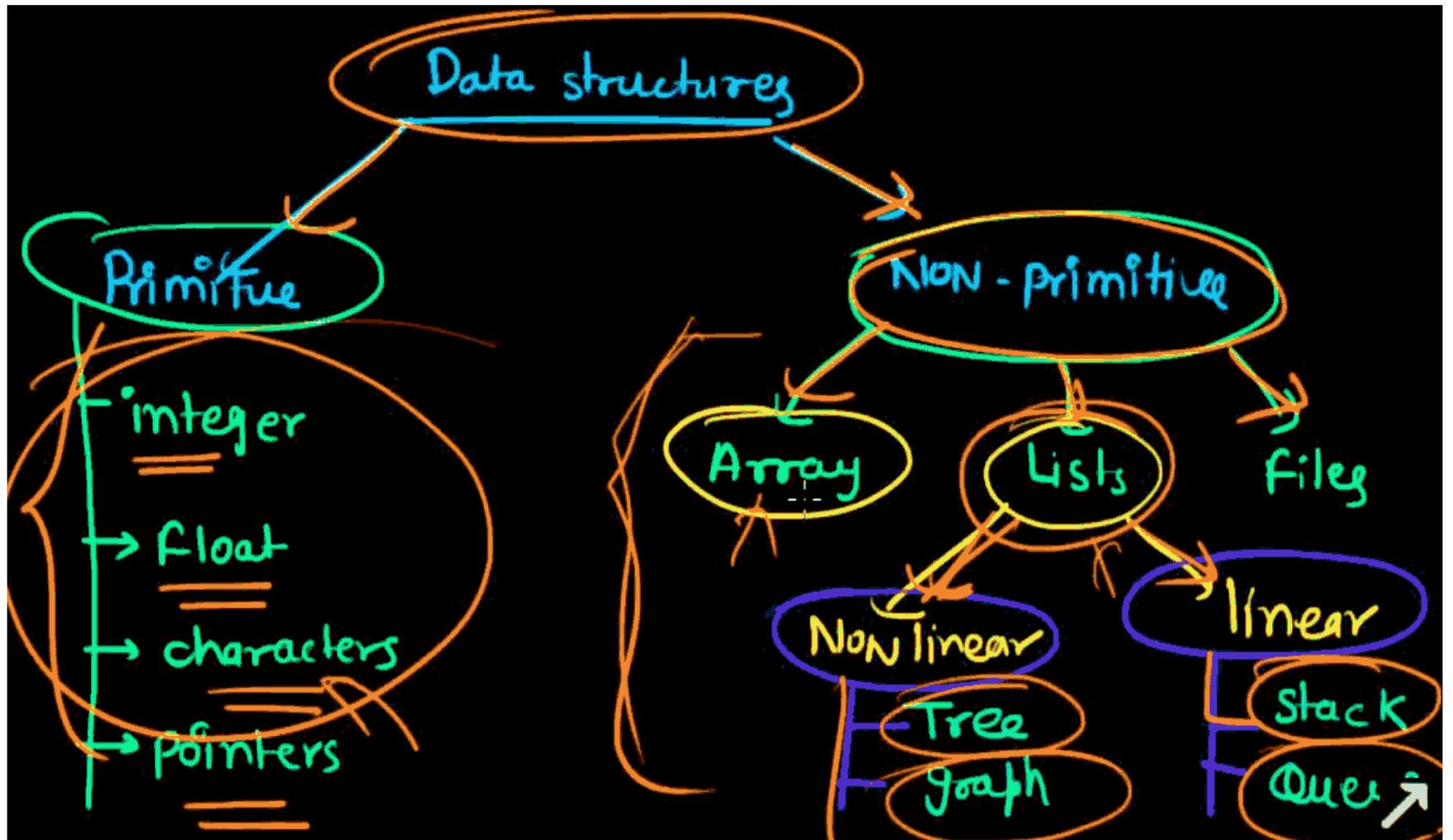
```
gatcttttta tttaaagat ctctttatta gatctottat taggatcatg atcctctgtg  
gataagtgat tattcacatg gcagatcata taattaagga ggatcgtttg ttgtgagtga  
coggtgatcg tattgcgat aagctgggat ctaaatggca tgttatgca agtcactcgg  
cagaaacaag gttgttatgt ggatctctac tggttttacc ctgcttttaa gcatagtatt  
acacattcgt tcgcccgatc tttgagctaa ttagagtaaa ttaatccaat ctttgacca
```

001010100101010101010010010010101000010010010100....



Evolution of the Desk

1980





## Digital Data Must Be ...

- **Encoded** (e.g. 01001001  $\leftrightarrow$  )

- **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

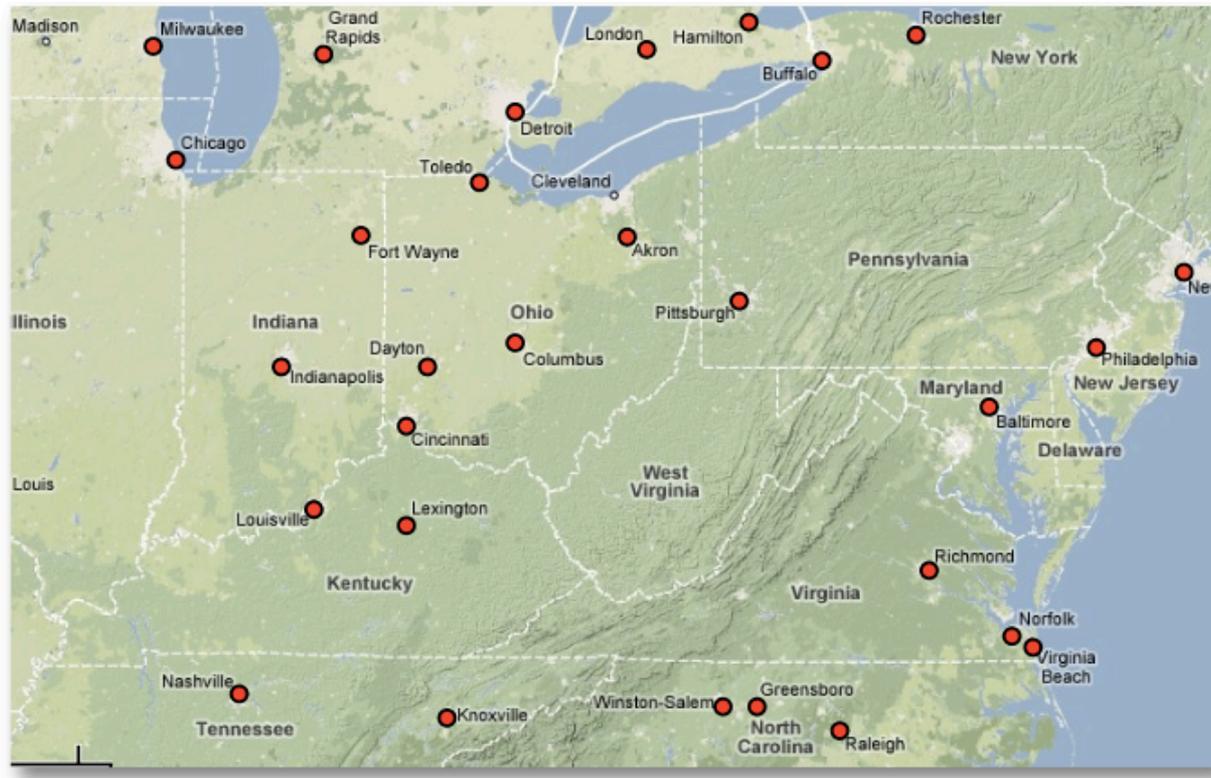
How do we organize information so that we can find, update, add, and delete portions of it efficiently?

# 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?

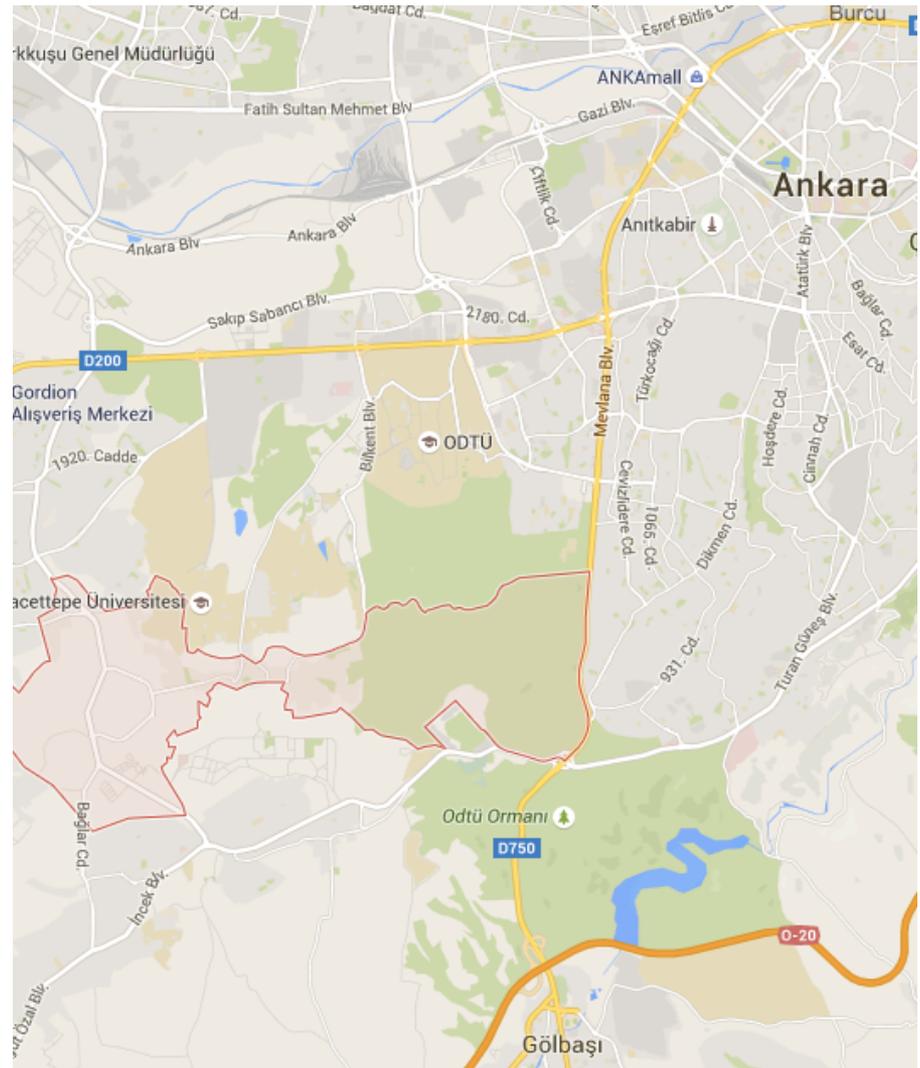
- *Find nearest point to me...*

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

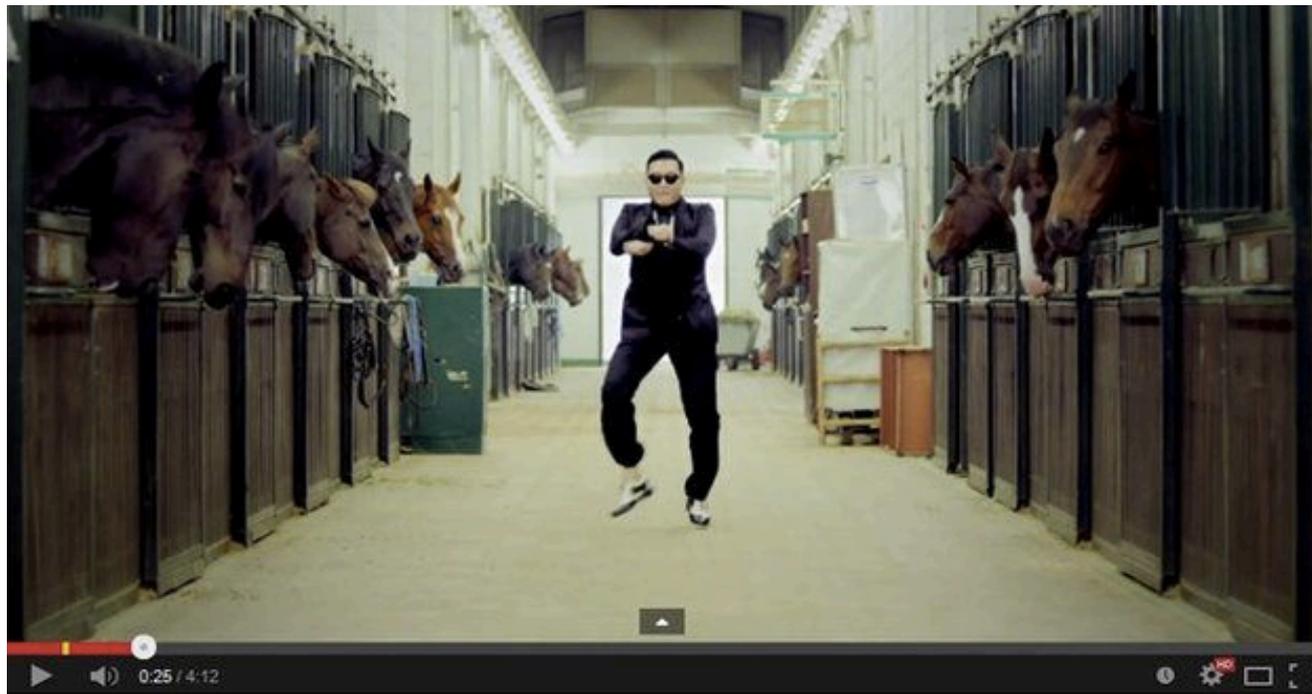


```
int userViewCount (int
current_count)
{
    int new_count;
    new_count =current_count + 1;
    return new_count;
}
```

Will this implementation work all the time?

# How will you count user views on YouTube?

%99.9 times yes.



PSY - GANGNAM STYLE (강남스타일) M/V



officialpsy

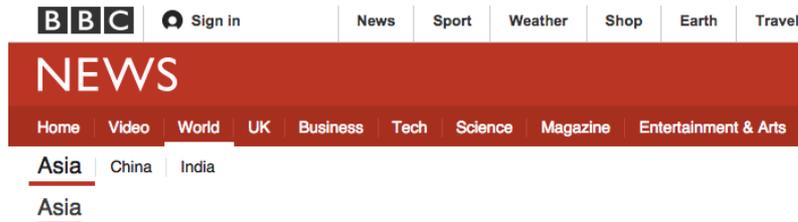
Subscribe 7,605,627

2,153,880,168

+ Add to Share ... More

8,781,922 1,142,528

# How will you count user views on YouTube?



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

4 December 2014 | Asia



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



## 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



# Floating Point Representation

## Format of Floating points IEEE754

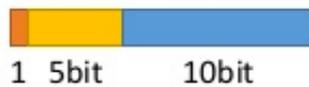
64bit = double, double precision



32bit = float, single precision



16bit = half, half precision



Signed bit

Exponent

Significand

# Floating Point Representation

| Nvidia Tesla Workstation GPU Performance Comparison |              |             |             |
|---|--------------|-------------|-------------|
|   | P100         | M40         | K40         |
| Architecture  | Pascal       | Maxwell     | Kepler      |
| Double Precision (FP64)                             | 5.3 Tflop/s  | 0.2 Tflop/s | 1.4 Tflop/s |
| Single Precision (FP32)                             | 10.6 Tflop/s | 7 Tflop/s   | 4.3 Tflop/s |
| Half Precision (FP16)                               | 21.1 Tflop/s | N/A         | N/A         |
| Memory Bandwidth                                    | 720GB/s      | 288GB/s     | 288GB/s     |
| Memory Size   | 16GB         | 12GB / 24GB | 12GB        |
| Release Date  | 2016         | Nov-15      | Nov-13      |

# 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





# 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.

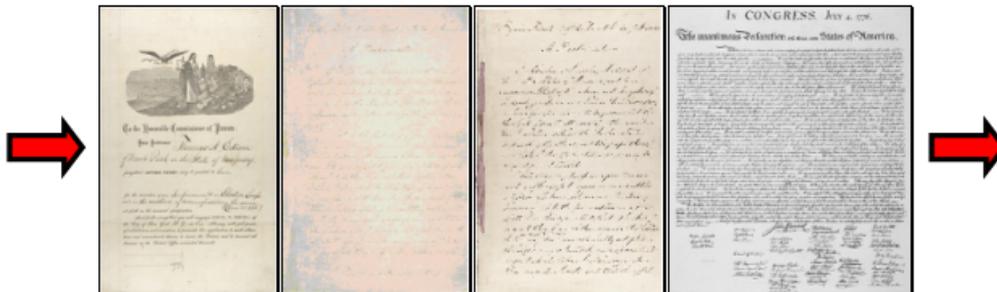
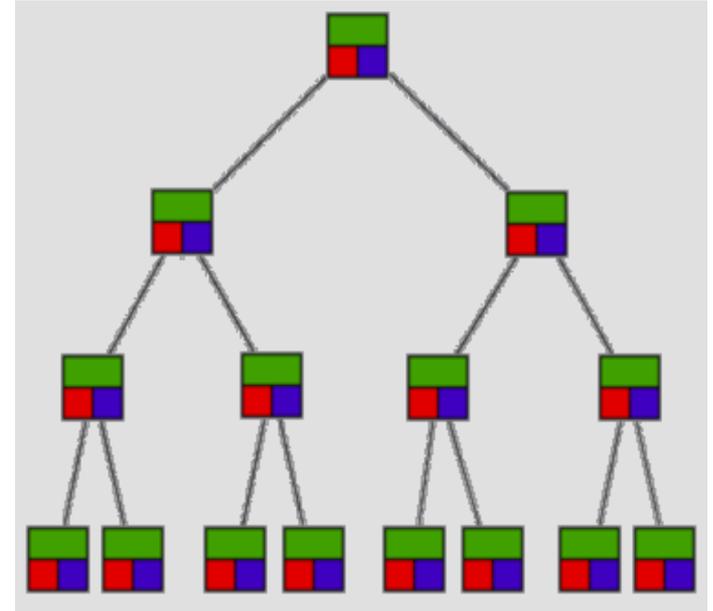
# Why so many?

- 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!*