

**CS 16**  
**Introduction to Algorithms**  
**and Data Structures**

with your host

**Roberto Tamassia**

# Why you should take CS 16

## Learn:

- All about efficient algorithms and data structures:
  - analysis
  - design
- How to implement algorithms and data structures in *Java*
- How to visualize data structures and algorithms

## Because:

- You're tired of code that looks like

```
public class CS015 extends GP.Confusion {}
```
- You have to if you want to get a computer science degree.
- You're cool like that.
- *Roberto* is cool like that.

# The CS 16 TA Team

## Grad TA:

Don “Blah” Blaheta

## Head TAs:

Mark “the Suspect” Handy

Benoit “not ‘Ben-oyt’” Hudson

## Ugrad TAs:

Ryan “Sleepy” Baker

Mike “Bubba” Boilen

Ming-En “Happy” Cho

Keith “Dribbs” Dreibelbis

Mark “Grumpy” Faktorovich

Seth “Sneezy” Landy

Phil “Howling Mad Cthulhu” Levis

Saul “Doc” Nadler

My Lan “Bashful” Payson

Soren “Kierkegaard” Spies

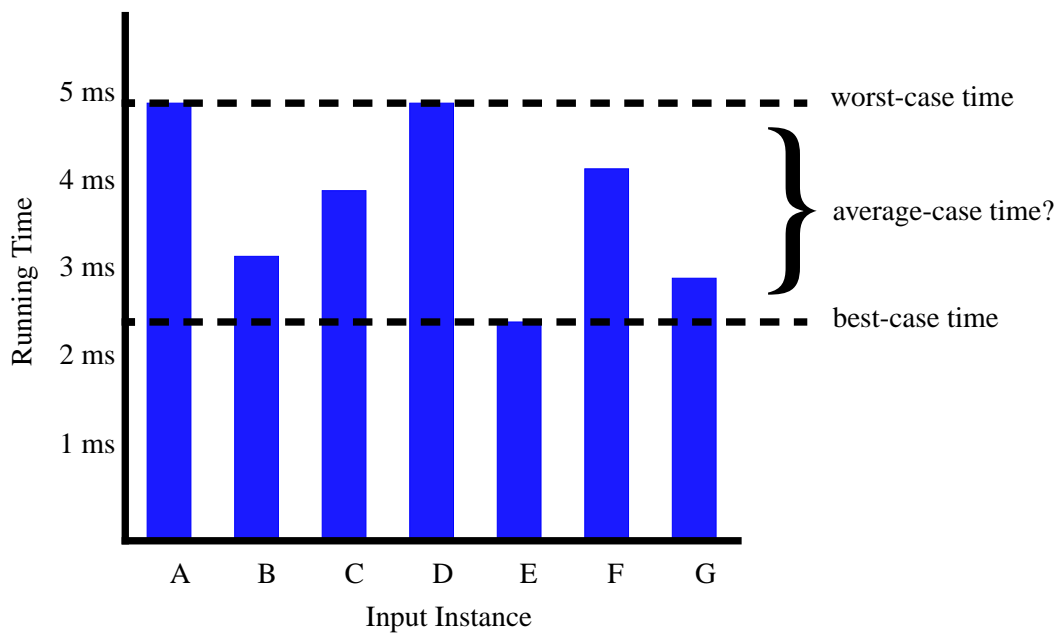
Greg “Laconic” Slovacek

Keith “Shaft” Schmidt

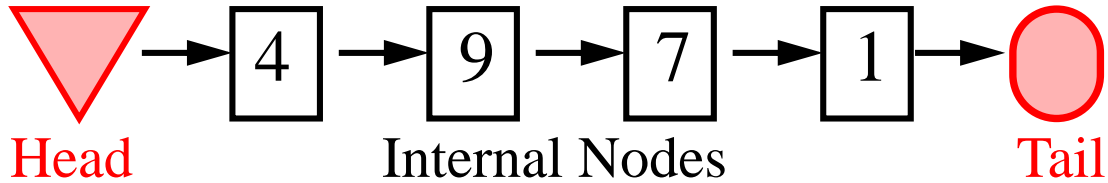
# Topics

- Analysis of Algorithms

## *BIG O*



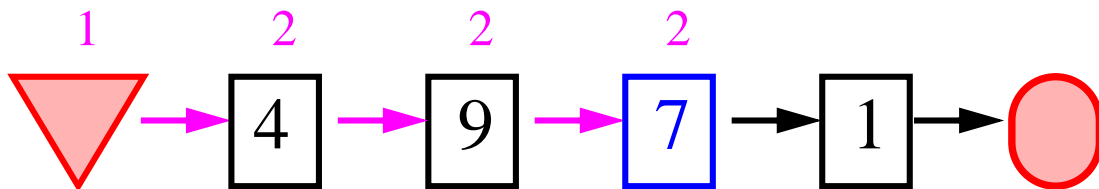
# Example: Search in a List



- Counting instructions:

Search(7);

# of instructions executed:

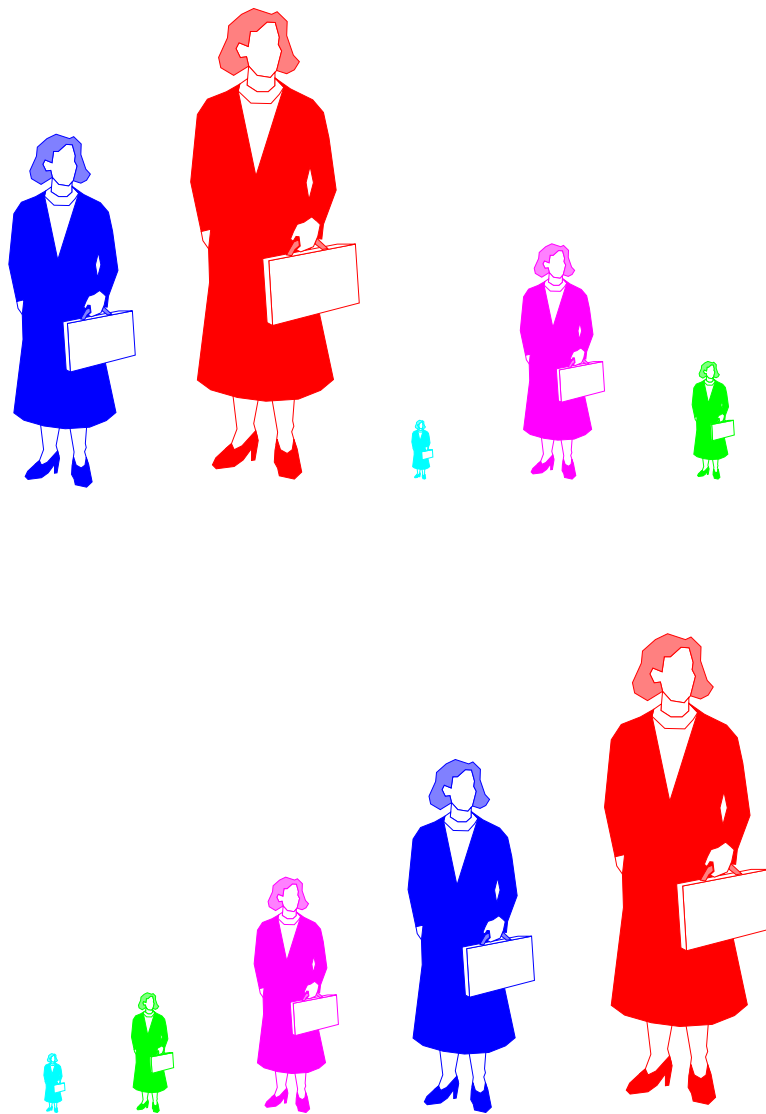


$$1 + 2 + 2 + 2 = 7$$

Finding an element in the third position requires 7 instructions. More generally, finding an element in the  $N$ th position requires  $2N + 1$  instructions.

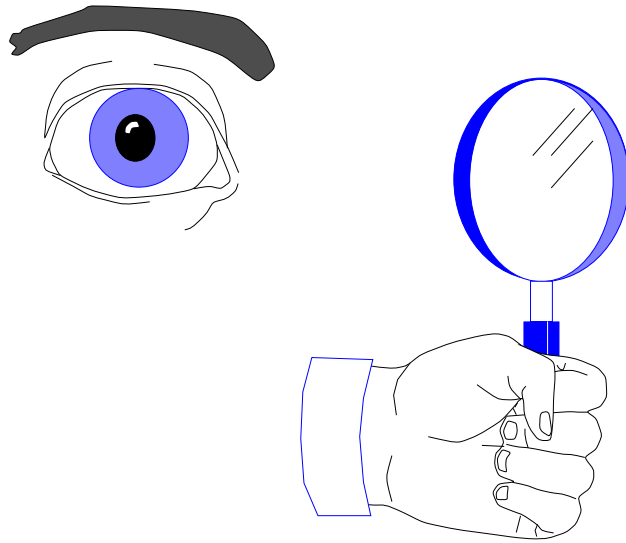
# Topics

- Sorting

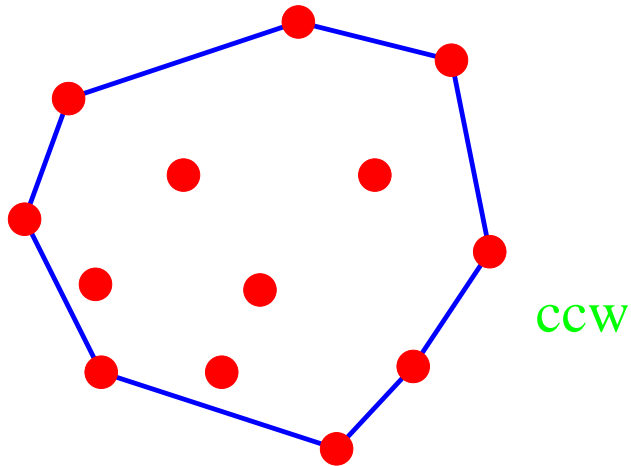


# Topics

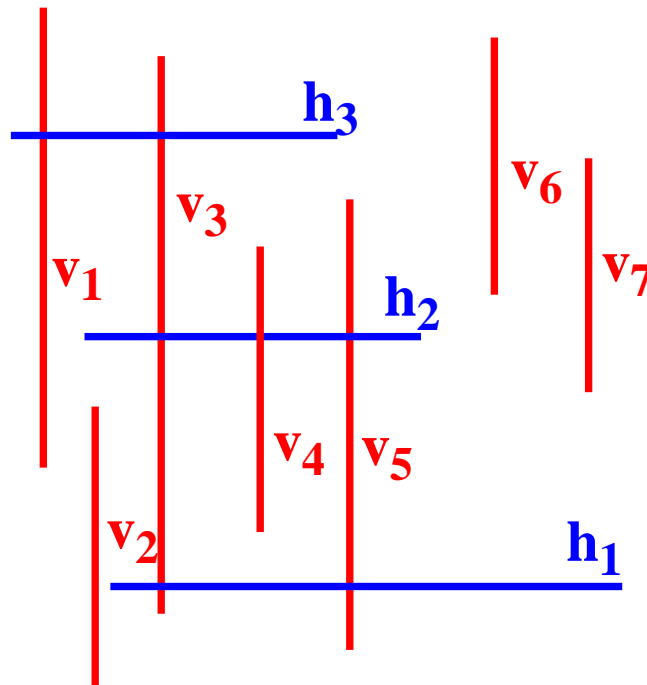
- Searching



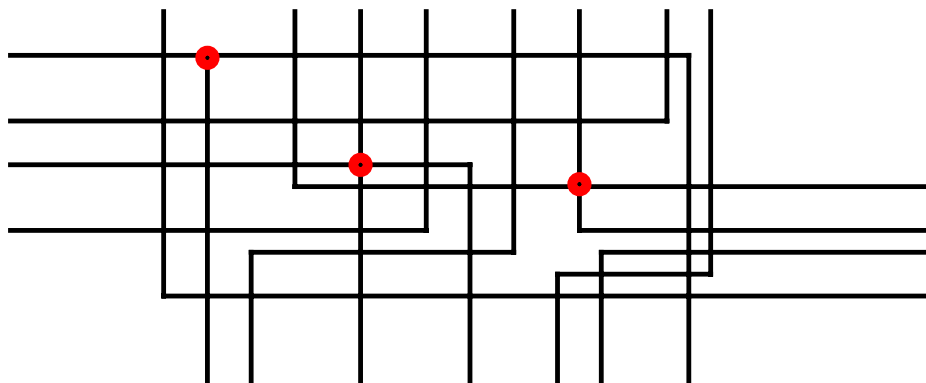
- Geometry



# Example: Intersection of Segments



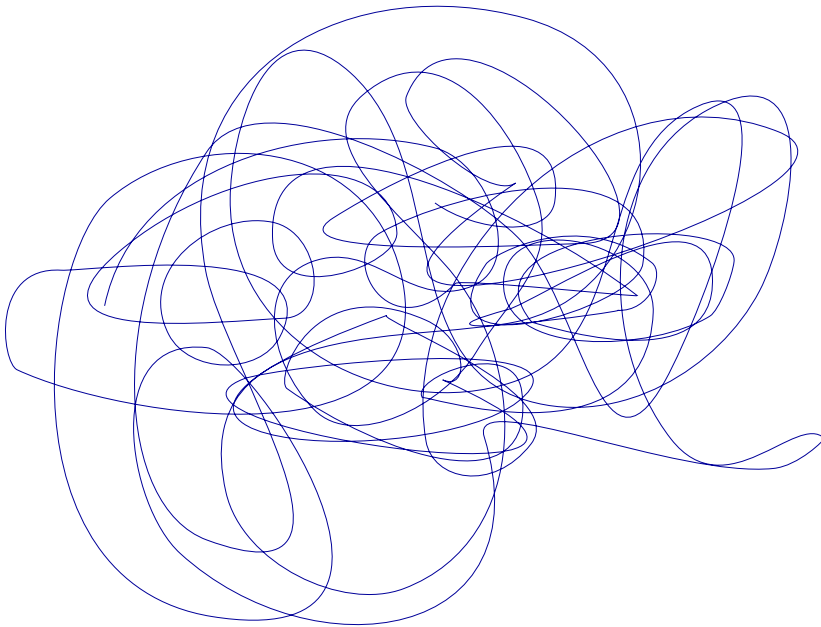
## Application to Circuit Layout



# Topics

- Text Processing
  - String Searching
  - Data Compression
  - Tries

There's my string!

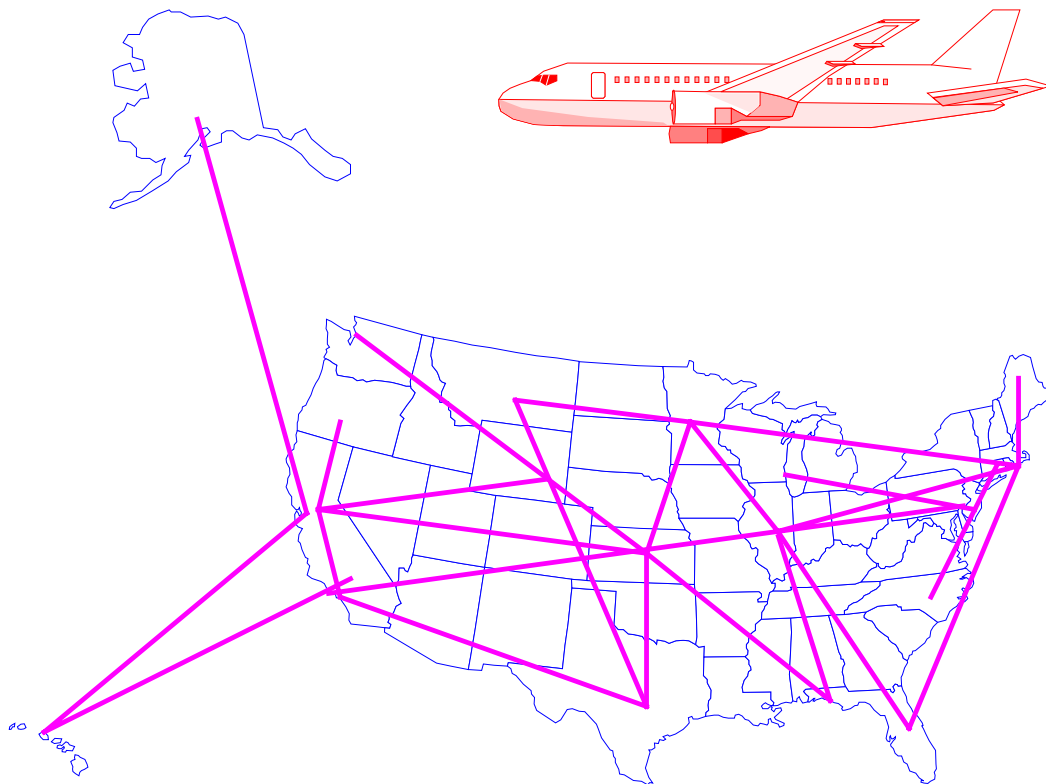


No! We're talking about:

$$x(i) = t[i] \cdot b^{M-1} + t[i+1] \cdot b^{M-2} + \dots + t[i+M-1]$$

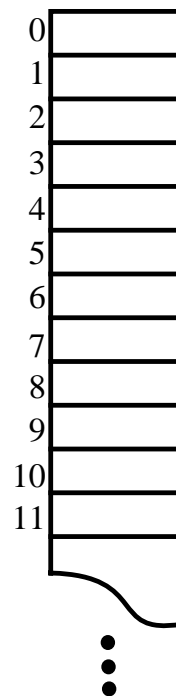
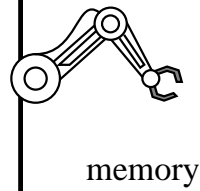
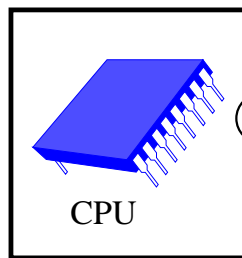
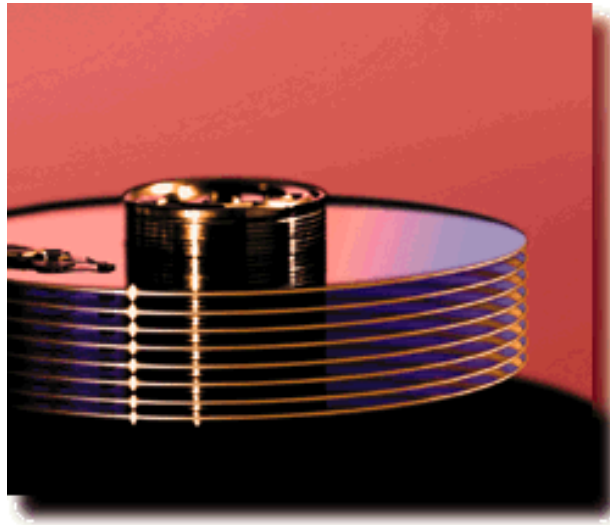
# Topics

- Graphs



# Topics

- Advanced Topics  
(e.g., Disks and Caches)



# Prerequisites

- **CS 15** *OR*
- **CS 4** *AND* **CS 5**
- For special cases, see Roberto at the end of class.
- elementary math
- Programming experience in **Java**: classes, objects, methods, references, ...

```
public class ArrayMaxProgram {  
    // test program for an algorithm that finds the maximum  
    // element in an array  
    ...  
    public static void main(String args[]) {  
        // testing method called when the program is executed  
        int[] num = { 10, 15, 3, 5, 56, 107, 22, 16, 85 };  
        int n = num.length;  
        System.out.print("Array:");  
        for (int i=0; i < n; i++)  
            System.out.print(" " + num[i]); // prints one element  
        System.out.println(".");  
        System.out.println("The maximum element is " +  
arrayMax(num,n) + ".");  
    }  
}
```

# Organization

- Lectures – attendance is important: not all of the course material is in the book
- 2 Exams
- 10 Homeworks
- 5 Mini-Programs
- 5 Programs
- Help Sessions – homeworks (mini-programs), programs
- Office Hours

## Final Grade

- Programs: 50%
- Homework: 30%
- Exams: 20%
- Special interest and class participation contributes to the determination of the final letter grade.

# Reading Material

## Required Reading:

- Michael T. Goodrich and [Roberto Tamassia](#). *Data Structures and Algorithms in Java*. New York: John Wiley & Sons: 1997.
- Additionally, some material not covered in the book will be handed out in lecture slide packets available in class.

## Today's Handouts:

- CS 16 Overview
- CS 16 Non-Collaboration Agreement

# Late Policy

## Programs:

- *Normal* due date and *late credits*
- Each student has 120 late credits (each credit extends the deadline by one hour)
- Grade declines over a period of 5 days
- Grade becomes **0** after 5 days
- Additional late credits may be assigned for medical reasons or emergencies.

## Homeworks:

- Only *one* due date
- *No* credit for late handin
- Worst homework dropped

# Collaboration Policy

- Limited collaboration for homeworks
- *No* collaboration on programs, mini-programs, or exams