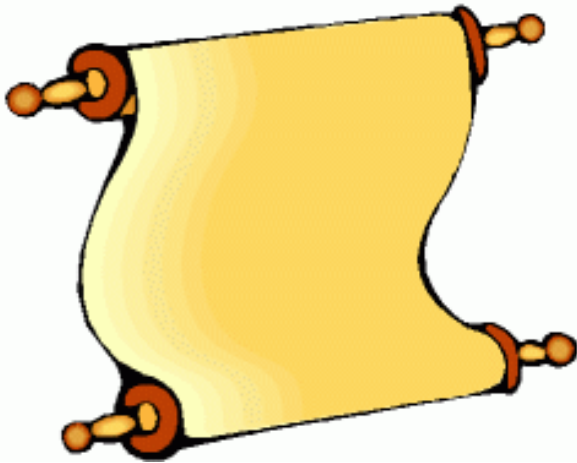
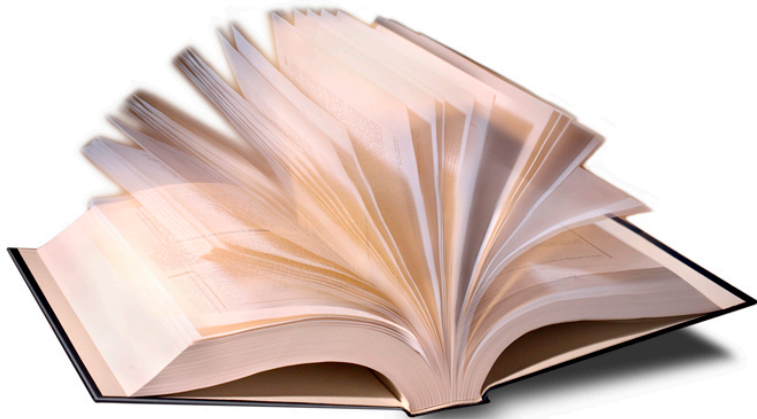


Data Structures and Algorithms

CS 110

Physical Data Structures



Which of these is the best?

Data Structures in a Computer

- We can program different ways of **organizing** data inside of the computer
- Each different way of organizing the data has various tradeoffs
 - What operations are supported? (e.g. add, delete, find, etc.)
 - How efficient are these operations?
 - How much overhead is involved in organizing the data?
- These issues are discussed at length in the Data Structures course

Data Structures in Processing

- Data structures can be implemented in Processing as classes
- What would the field of the class be in this case?
- What would the methods be?

Built-in Collection Classes

- ArrayList
 - A built-in object that stores and manages an *arbitrary* number of data items of any type (Objects).
 - Objects in an ArrayList are access by **index** [0..size-1]
- HashMap
 - A built-in object that stores and manages an *arbitrary* number of data items of any type (Objects).
 - Objects in a HashMap are access by a **key**, which can be another Object, frequently a String.

ArrayList

- Constructors

```
ArrayList<Object> lst1 = new ArrayList();  
ArrayList<Object> lst2 = new ArrayList(initialSize);
```

- Fields

- Methods

<code>size()</code>	<code>// Returns the num of items held.</code>
<code>add(Object o)</code>	<code>// Appends o to end.</code>
<code>add(int idx, Object o)</code>	<code>// Inserts o at pos idx.</code>
<code>remove(int idx)</code>	<code>// Removes item at pos idx.</code>
<code>get(int idx)</code>	<code>// Gets items at idx. No removal.</code>
<code>set(int idx, Object o)</code>	<code>// Replaces item at idx with o.</code>
<code>clear()</code>	<code>// Removes all items.</code>
<code>isEmpty()</code>	<code>// true if empty.</code>

ArrayList Example – Box Dropper

```
// Box Dropper
ArrayList<Box> boxes = new ArrayList();
void setup() {
    size(500, 500);
}

void draw() {
    background(0);

    for (int i = boxes.size()-1; i>=0; i--) {
        boxes.get(i).update();
        boxes.get(i).draw();
        // Remove Box from ArrayList if below sketch
        if (boxes.get(i).y > height) {
            boxes.remove(i);
            println(boxes.size() + " boxes remaining");
        }
    }
}
```

```
void mousePressed() {
    Box b = new Box(mouseX, mouseY);
    boxes.add( b );
    println( boxes.size() + " boxes in ArrayList" );
}

class Box {
    float x, y, v;

    Box(float tx, float ty) {
        x = tx; // x position
        y = ty; // y position
        v = 0.0; // y velocity
    }

    void draw() {
        fill(200);
        rect(x, y, 20, 20);
    }

    void update() {
        y += v; // Physics
        v += 0.02;
    }
}
```

Cooler Array List Example



The Basic Idea

- Start with a single tile that contains the entire image
- On each frame
 - Choose random tile to start falling
 - Choose random tile to split into two new tiles
- At a high-level, how would we use the methods of the ArrayList to implement these two steps?

HashMap

- Constructors

```
HashMap<Object, Object> map1 = new HashMap();  
HashMap<Object, Object> map2 = new HashMap(initialCapacity);
```

- Fields

- Methods

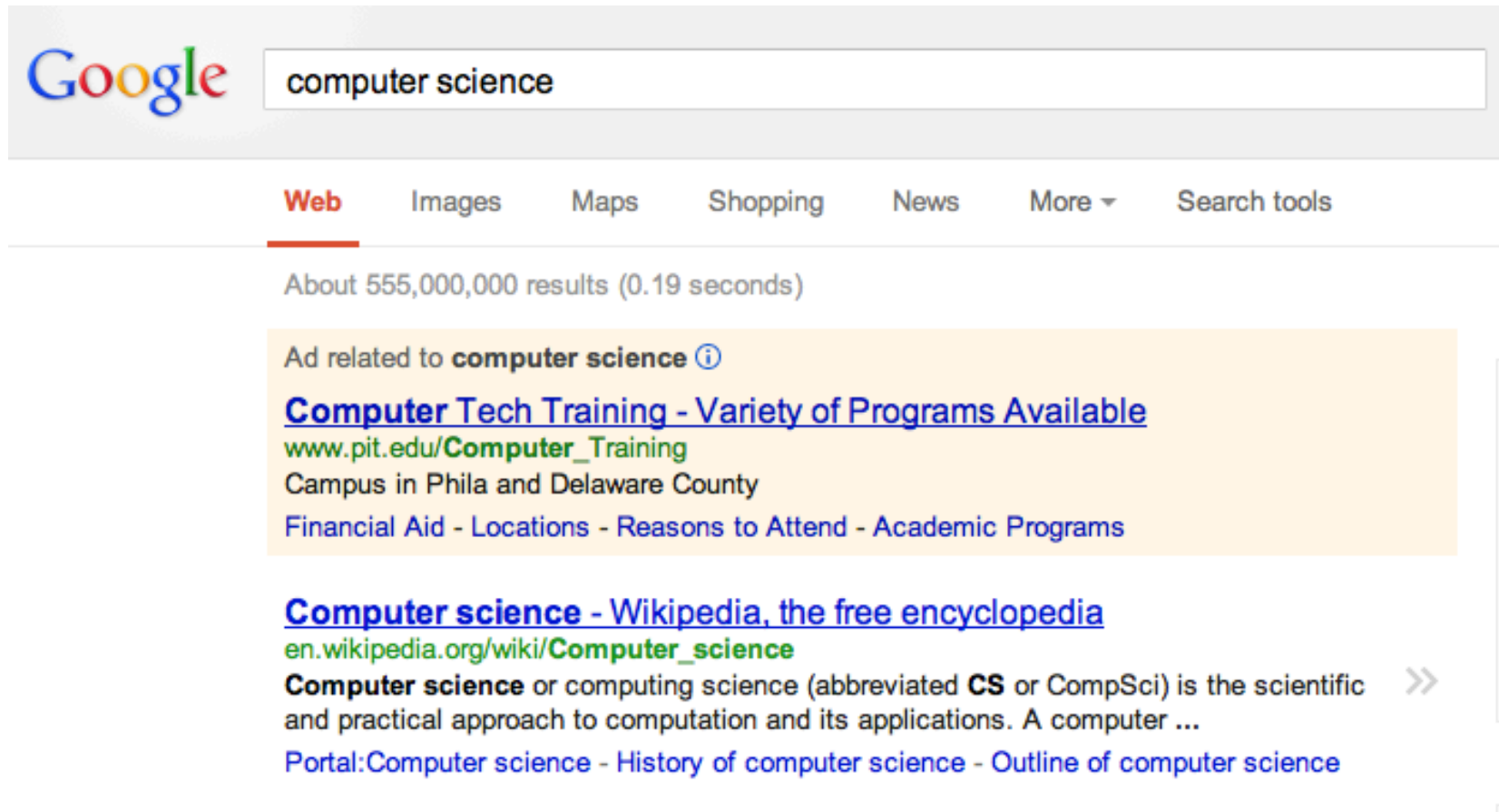
<code>size()</code>	<code>// Returns num of items held.</code>
<code>put(Object key, Object o)</code>	<code>// Puts o in map at key</code>
<code>remove(Object key)</code>	<code>// Remove Object at key</code>
<code>get(Object key)</code>	<code>// Get Object at key</code>
<code>containsKey(Object key)</code>	<code>// True if map contains key</code>
<code>containsValue(Object val)</code>	<code>// True if map contains val</code>
<code>clear()</code>	<code>// Removes all items.</code>
<code>isEmpty()</code>	<code>// true if empty.</code>

More Data Structures

Trees,
Heaps,
Graphs,
Linked Lists,
Queues,
Stacks,
etc.

http://en.wikipedia.org/wiki/List_of_data_structures

Imagine the Data Structures that Google Uses



The image is a screenshot of a Google search interface. At the top left is the Google logo. To its right is a search bar containing the text "computer science". Below the search bar is a horizontal navigation bar with links: "Web" (highlighted with a red underline), "Images", "Maps", "Shopping", "News", "More" (with a dropdown arrow), and "Search tools". Below the navigation bar, the search results are displayed. The first line of results says "About 555,000,000 results (0.19 seconds)". Below this is a yellow background box containing an advertisement. The ad is titled "Ad related to **computer science** ⓘ" and features the link "[Computer Tech Training - Variety of Programs Available](\"http://www.pit.edu/Computer_Training\")". Below the link is the URL "www.pit.edu/Computer_Training" and the text "Campus in Phila and Delaware County". At the bottom of the ad box are the links "Financial Aid - Locations - Reasons to Attend - Academic Programs". Below the yellow box is a search result from Wikipedia. The title is "[Computer science - Wikipedia, the free encyclopedia](\"http://en.wikipedia.org/wiki/Computer_science\")". Below the title is the URL "en.wikipedia.org/wiki/Computer_science". The main text of the result is "Computer science or computing science (abbreviated **CS** or CompSci) is the scientific and practical approach to computation and its applications. A computer ...". To the right of this text is a double right arrow "»". At the bottom of the result is a list of links: "Portal:Computer science - History of computer science - Outline of computer science".

Google

computer science

Web Images Maps Shopping News More ▾ Search tools

About 555,000,000 results (0.19 seconds)

Ad related to **computer science** ⓘ

[Computer Tech Training - Variety of Programs Available](http://www.pit.edu/Computer_Training)
www.pit.edu/Computer_Training
Campus in Phila and Delaware County
Financial Aid - Locations - Reasons to Attend - Academic Programs

[Computer science - Wikipedia, the free encyclopedia](http://en.wikipedia.org/wiki/Computer_science)
en.wikipedia.org/wiki/Computer_science
Computer science or computing science (abbreviated **CS** or CompSci) is the scientific and practical approach to computation and its applications. A computer ...
Portal:Computer science - History of computer science - Outline of computer science

Algorithm

- A well-defined set of instructions for solving a particular kind of problem.
- Algorithms exist for systematically solving many types of problems
 - Sorting
 - Searching
 - ...

Searching

This is a fundamentally important problem for a myriad of applications (from finding webpages to searching for fragments of DNA)

The problem:

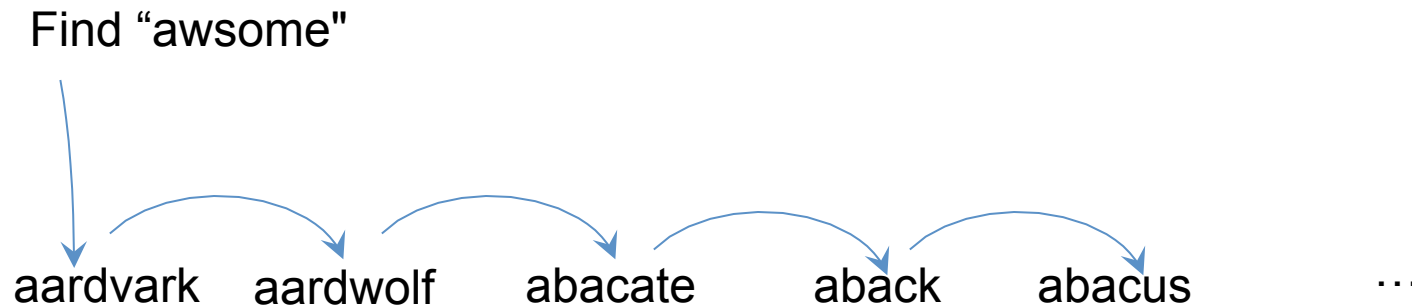
Given a collection of data, determine if a query is contained in that collection.

Motivating Example: Spellchecker!



Exhaustive (Linear) Search

- Systematically enumerate all possible values and compare to value being sought
- For an array, iterate from the beginning to the end, and test each item in the array



Binary Search

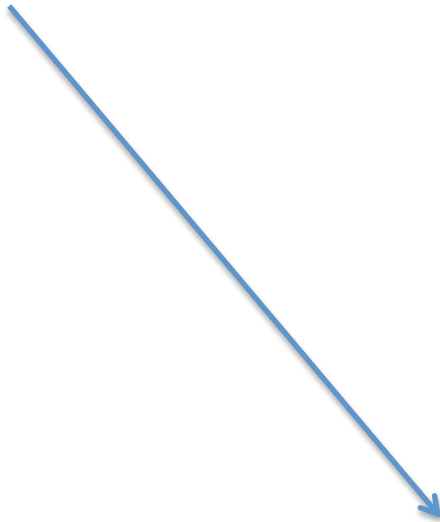
- Quickly find an item (val) in a sorted list.
- Recursive Procedure:

```
binarySearch(startIdx, stopIdx, data, query)
    if startIdx == stopIdx
        return data[startIdx] == query
    midPoint = (startIdx+stopIdx)/2
    if words[midPoint] >= query
        return binarySearch(startIdx, midPoint, data, query)
    else
        return binarySearch(midPoint+1, stopIdx, data, query)
```

The most efficient way to play "guess the number" ...

Binary Search Example

Find "awsome"



aardvark

...

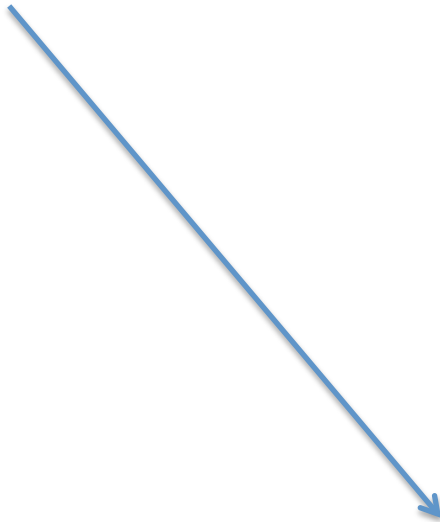
macabre

...

Zyzzogeton

Binary Search Example

Find "awesome"



aardvark

...

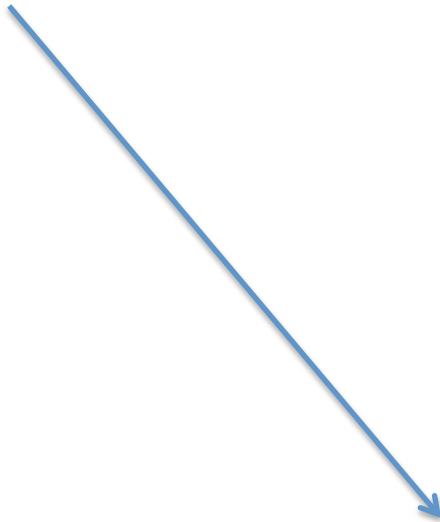
fable

...

macabre

Binary Search Example

Find "awesome"



aardvark

...

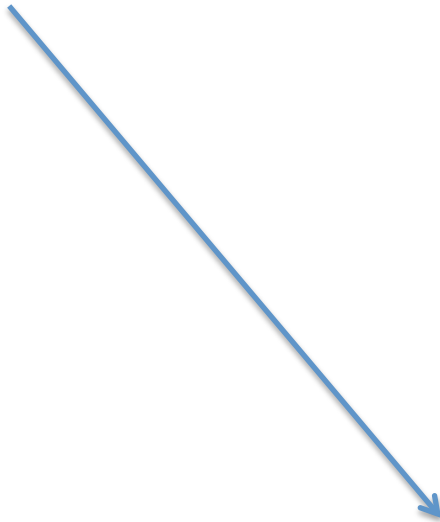
catfish

...

fable

Binary Search Example

Find "awesome"



aardvark

...

beetle

...

catfish

Binary Search Example

Find "awesome"

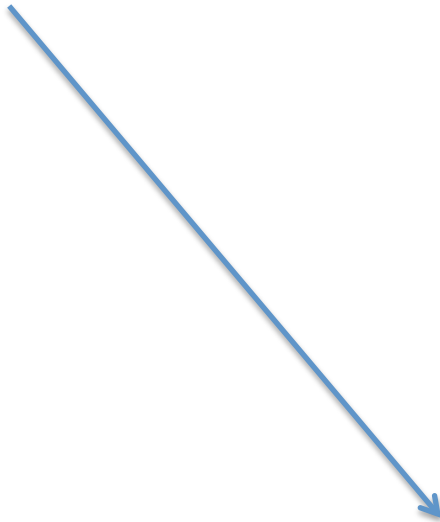
aardvark

...

awake

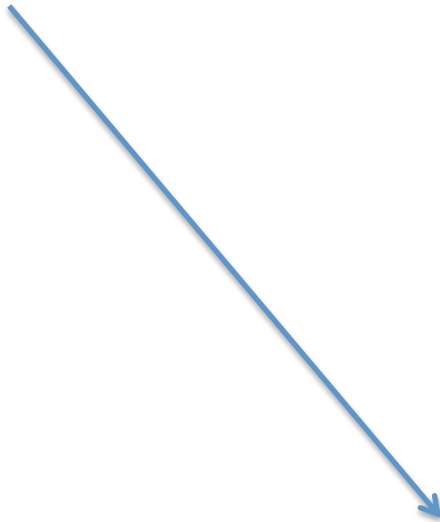
...

beetle



Binary Search Example

Find "awesome"



awaken

...

banjo

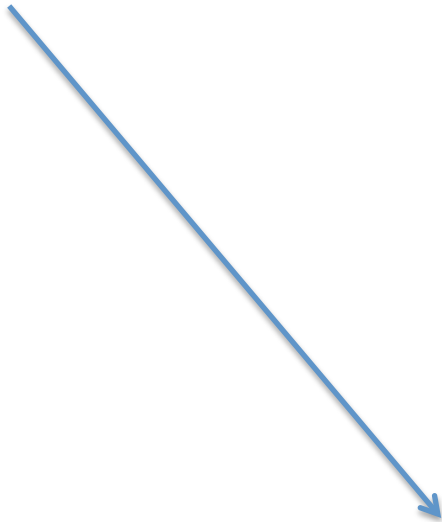
...

beetle

Repeat a few more times...

Binary Search Example

Find "awesome"



awry

Return false

Spell-Checking Midsummer Night's Dream

- Exhaustive Search: **385,554 milliseconds** to check the entire text
- Binary Search: **104 milliseconds** to check the entire text
- Speedup: **3,707 times!!!**

Forbes
com

Microsoft

DON'T GET FORCED. GET
Switch and get \$150/user ▶

In Pictures: Weird Job Interview Questions



"Can I Guess?"

Given the numbers 1 to 1,000, what is the minimum number of guesses needed to find a specific number if you are given the hint "higher" or "lower" for each guess you make?

Asked at Facebook

CLOSE



Sitcom/Dictator Game



<http://www.smalltime.com/Dictator>

Transitioning to Java in 1 Slide

Processing:

```
void setup() {  
    println("Hello World!");  
}
```

Java:

```
class test {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```