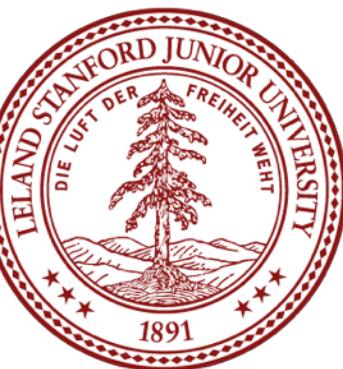


Sets and Maps

CS 106B

Programming Abstractions
Fall 2016
Stanford University
Computer Science Department

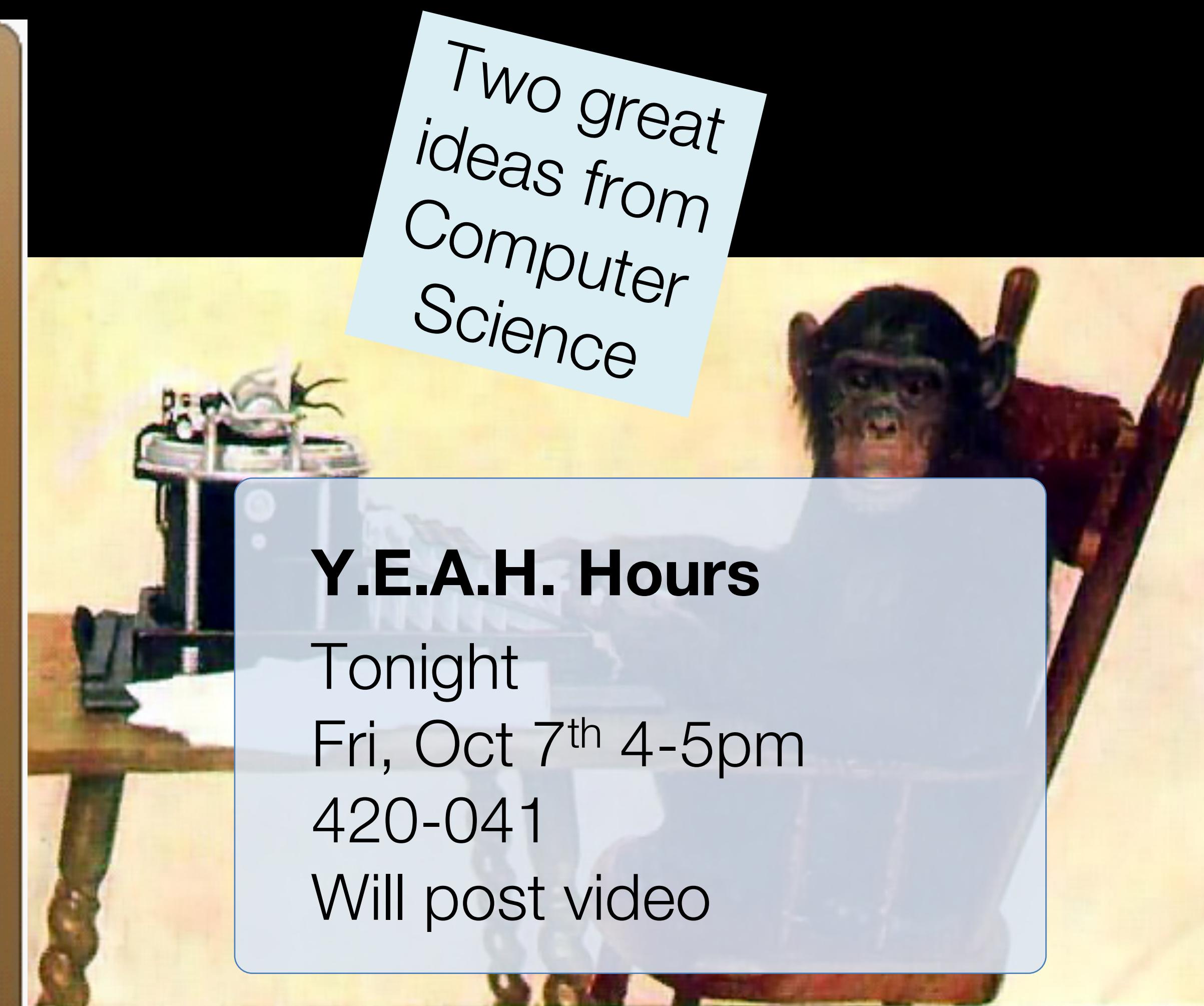


Announcements

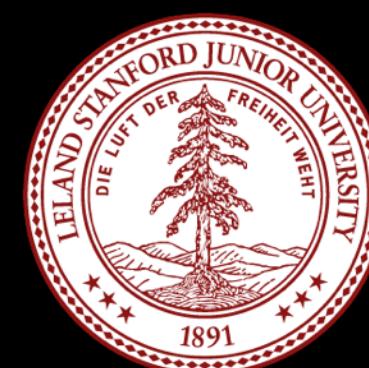
- ▶ Assignment 1 was due today.
- ▶ Assignment 2 goes out today.



Assignment 2: Serafini



Due next Saturday



Interesting Puzzle

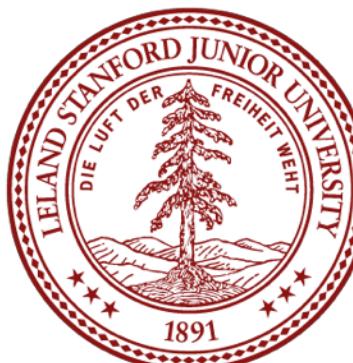
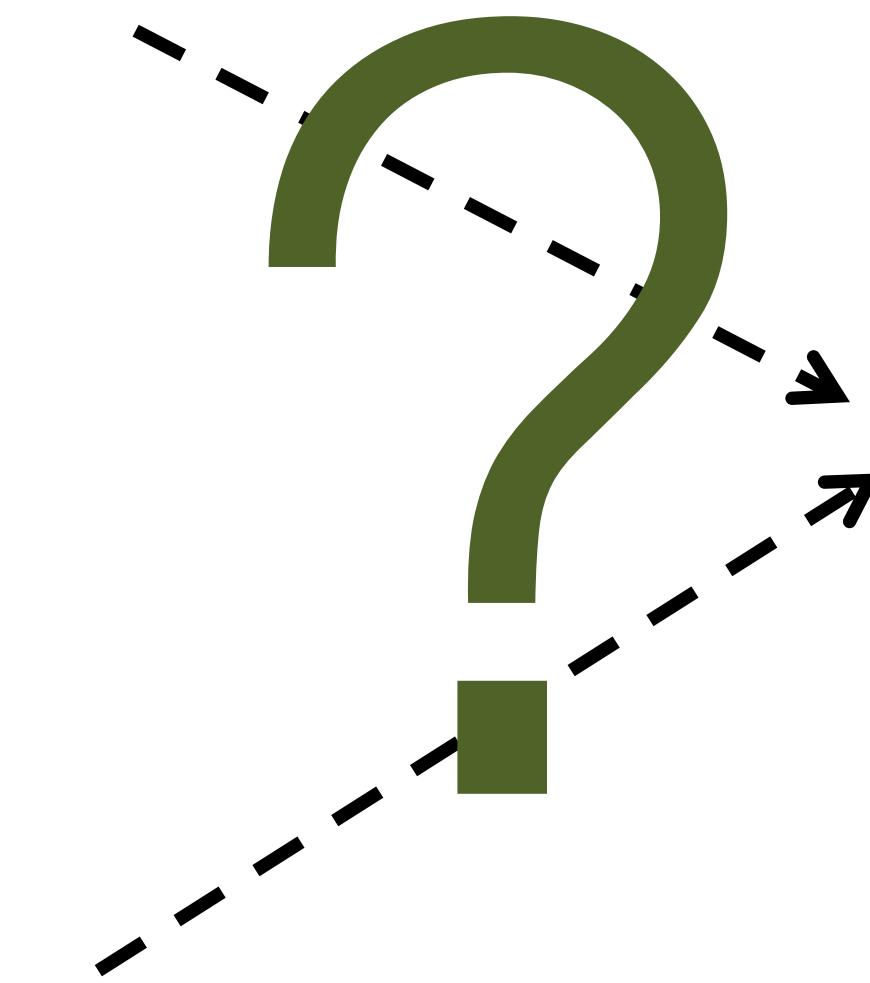
Counterfeiter



User

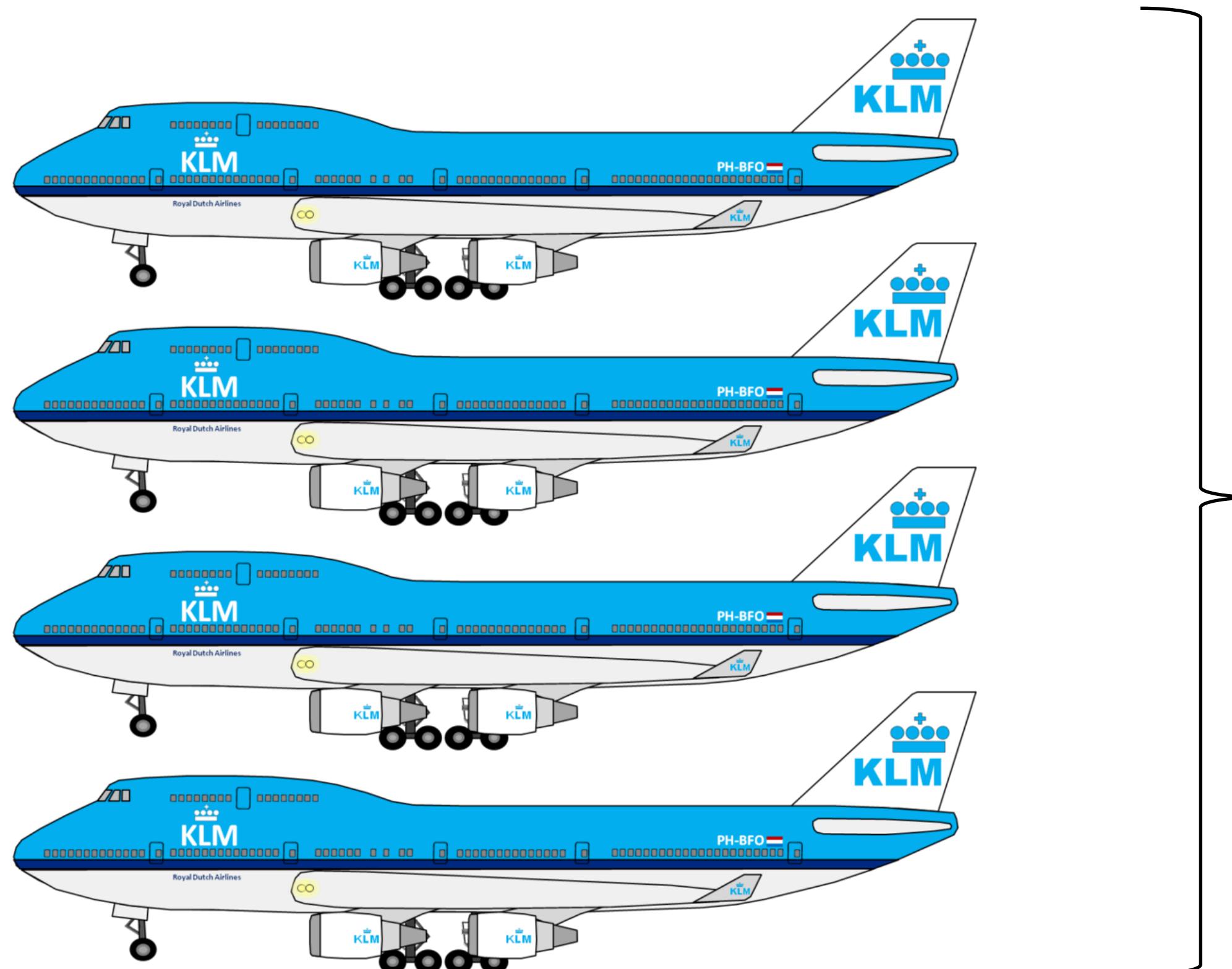


You (Distributor)



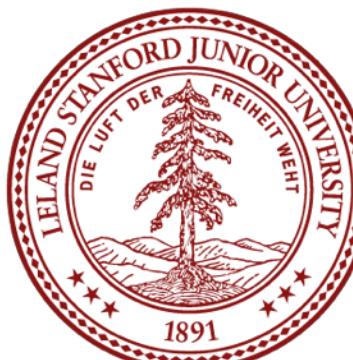
Fake Medicine is a Problem

700,000 deaths from fake malaria and tuberculosis drugs in 2013 [1]



Equivalent of
this many
crashes per
day

[1] <http://www.un.org/africarenewal/magazine/may-2013/counterfeit-drugs-raise-africa%E2%80%99s-temperature>



Abstract Data Types

Vector

Grid

Map

Stack

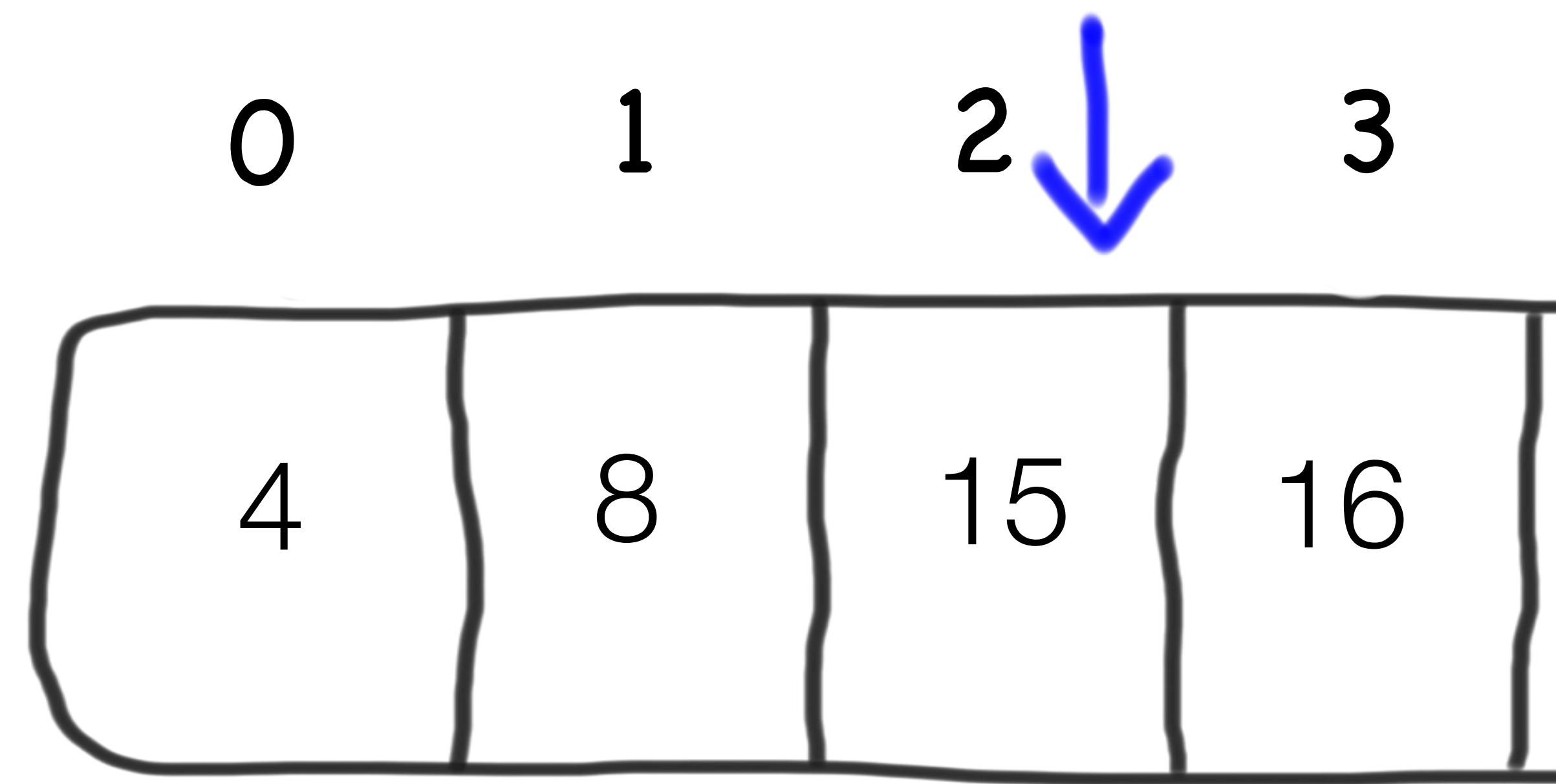
Queue

Set



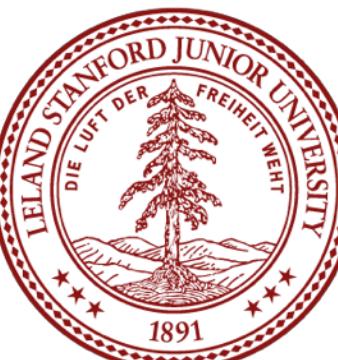
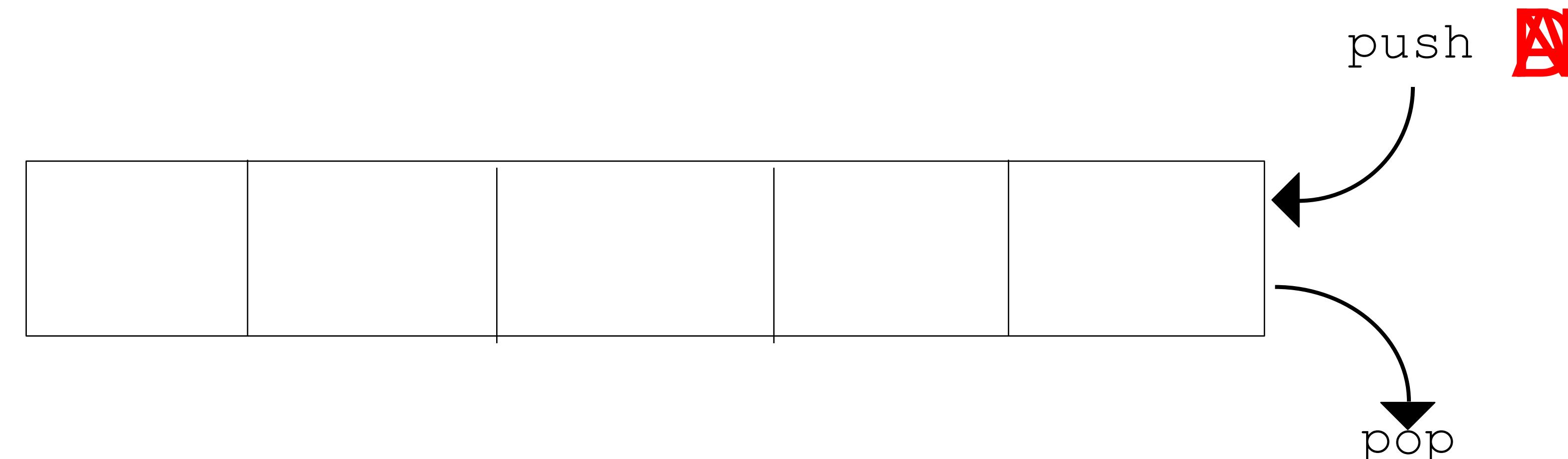
Vectors

- ▶ Can grow to any size you need.
- ▶ Supports random access.

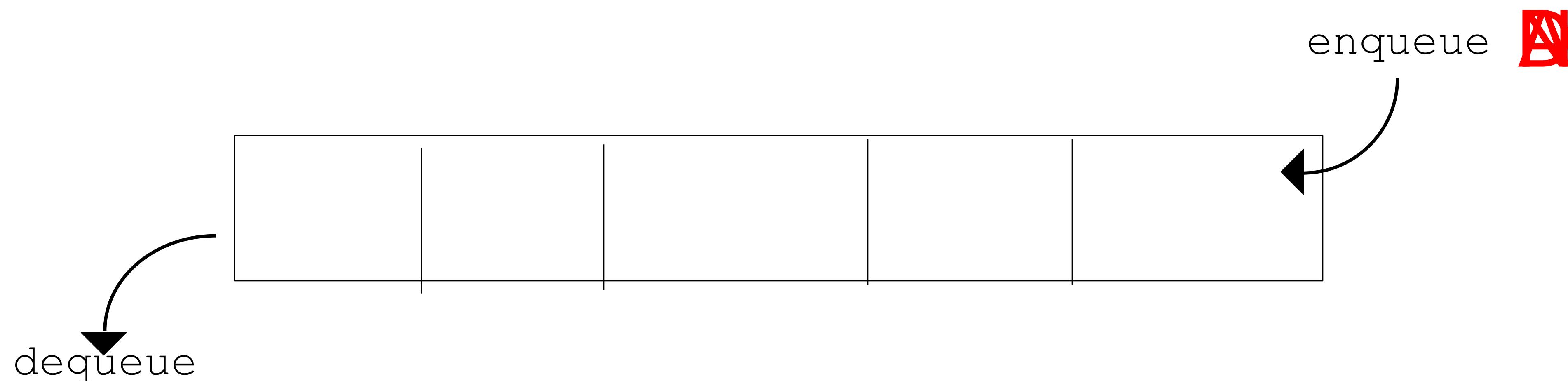


Stack

- ▶ Last in, First Out (so rude)
- ▶ Slightly faster than vector.
- ▶ Great style.



- ▶ First in, First Out (very fair)
- ▶ Slightly faster than vector.
- ▶ Great style.



Abstract Data Types

Vector

Grid

Map

Stack

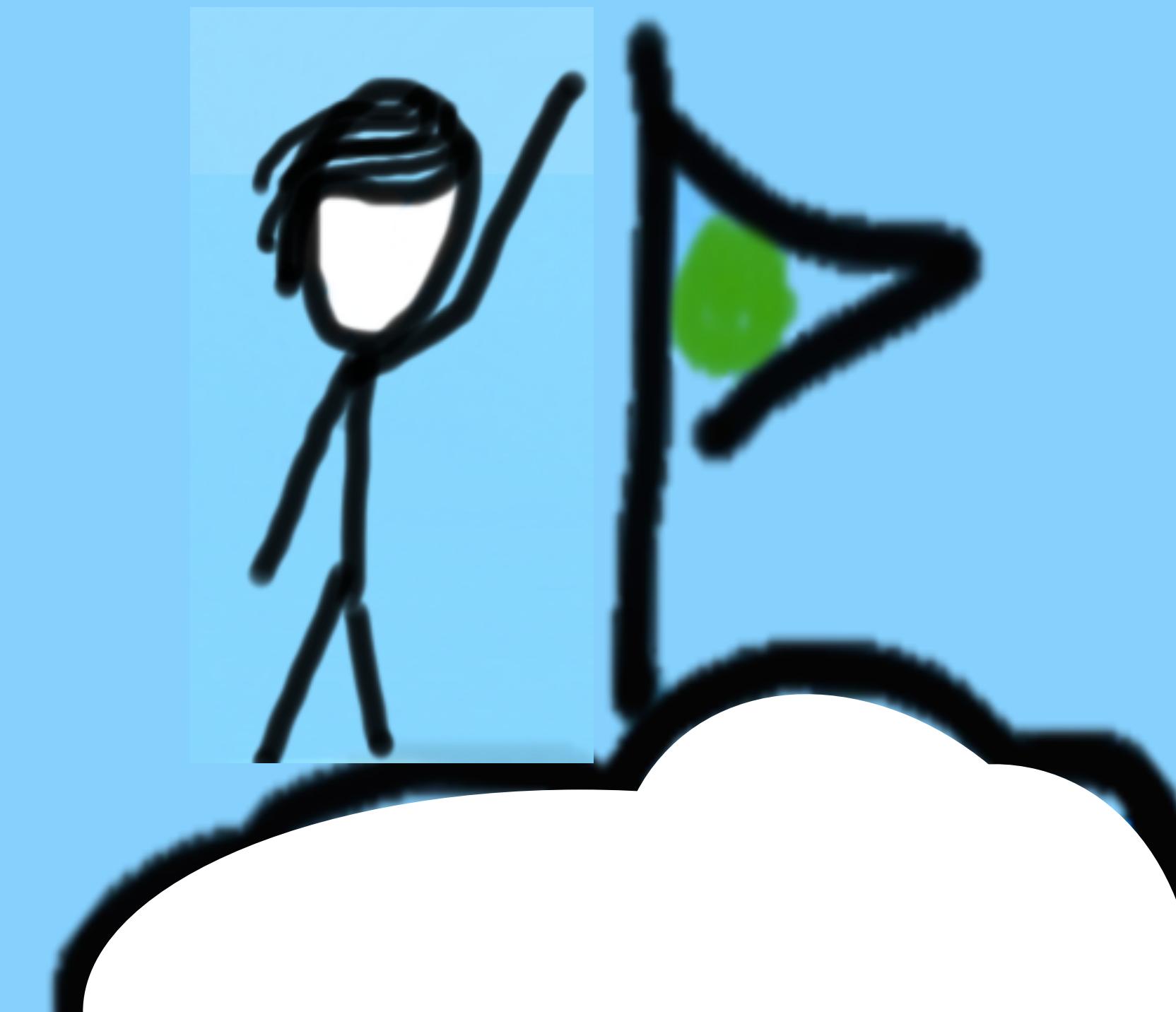
Queue

Set

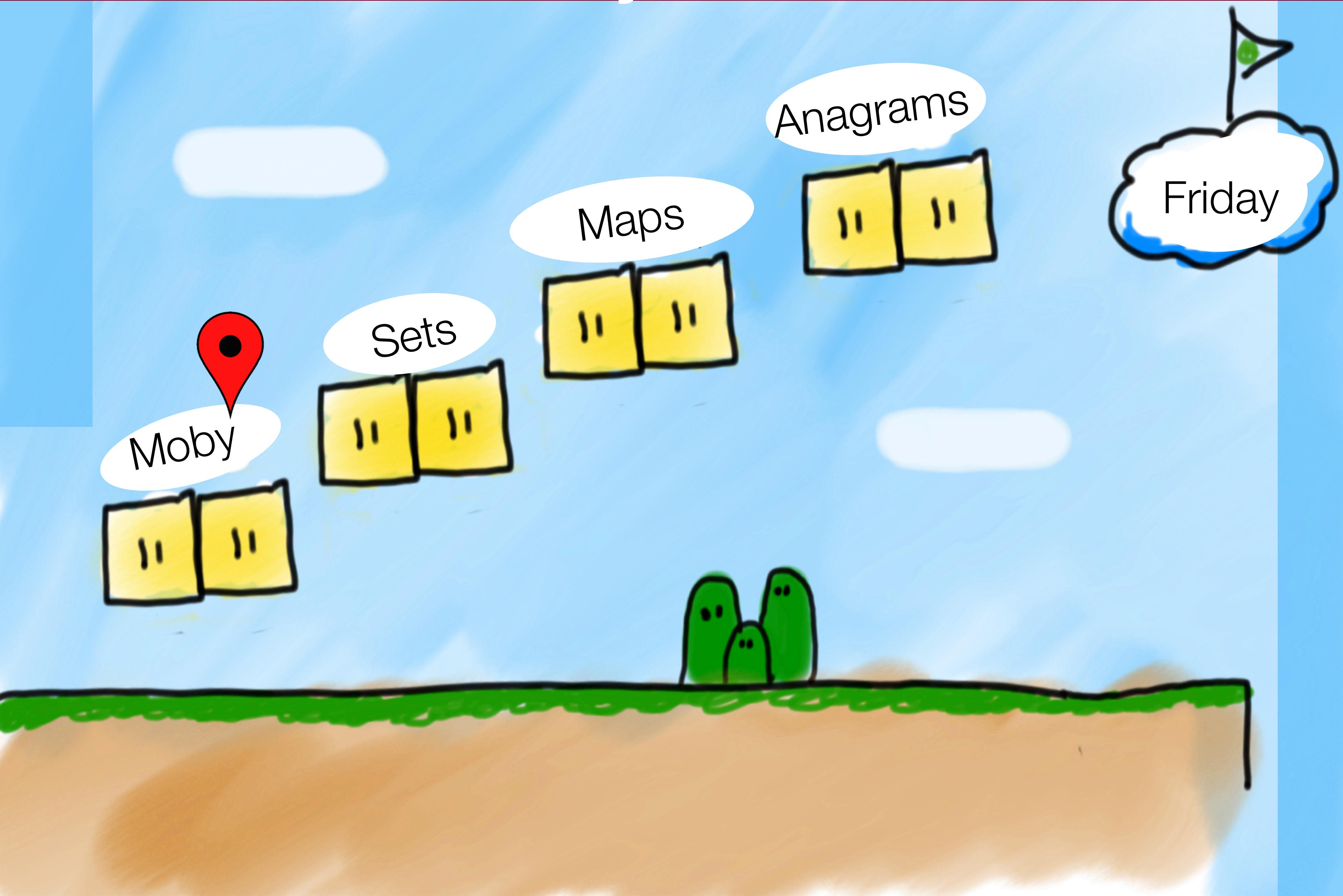


Today's Goal

1. Learn how to use Sets
2. Learn how to use Maps
3. Be prepared for Assn 2



Today's Plan



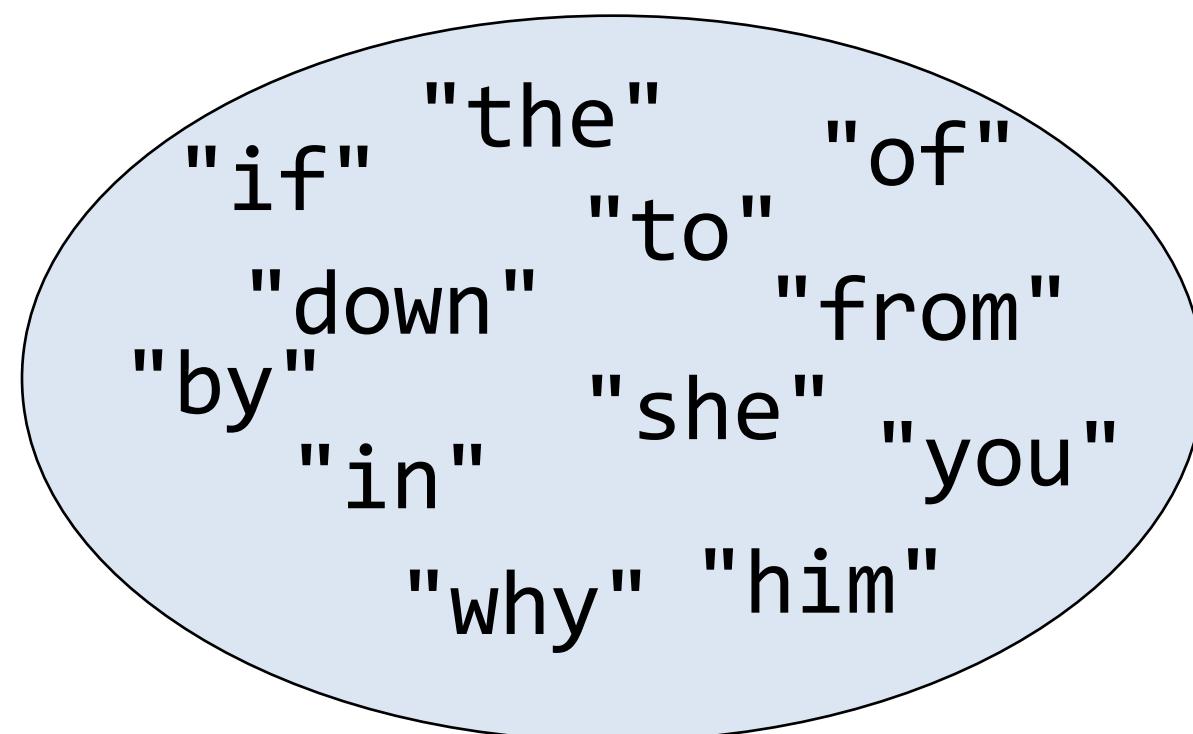
Problem #1: Count Unique Words



Sets and Maps

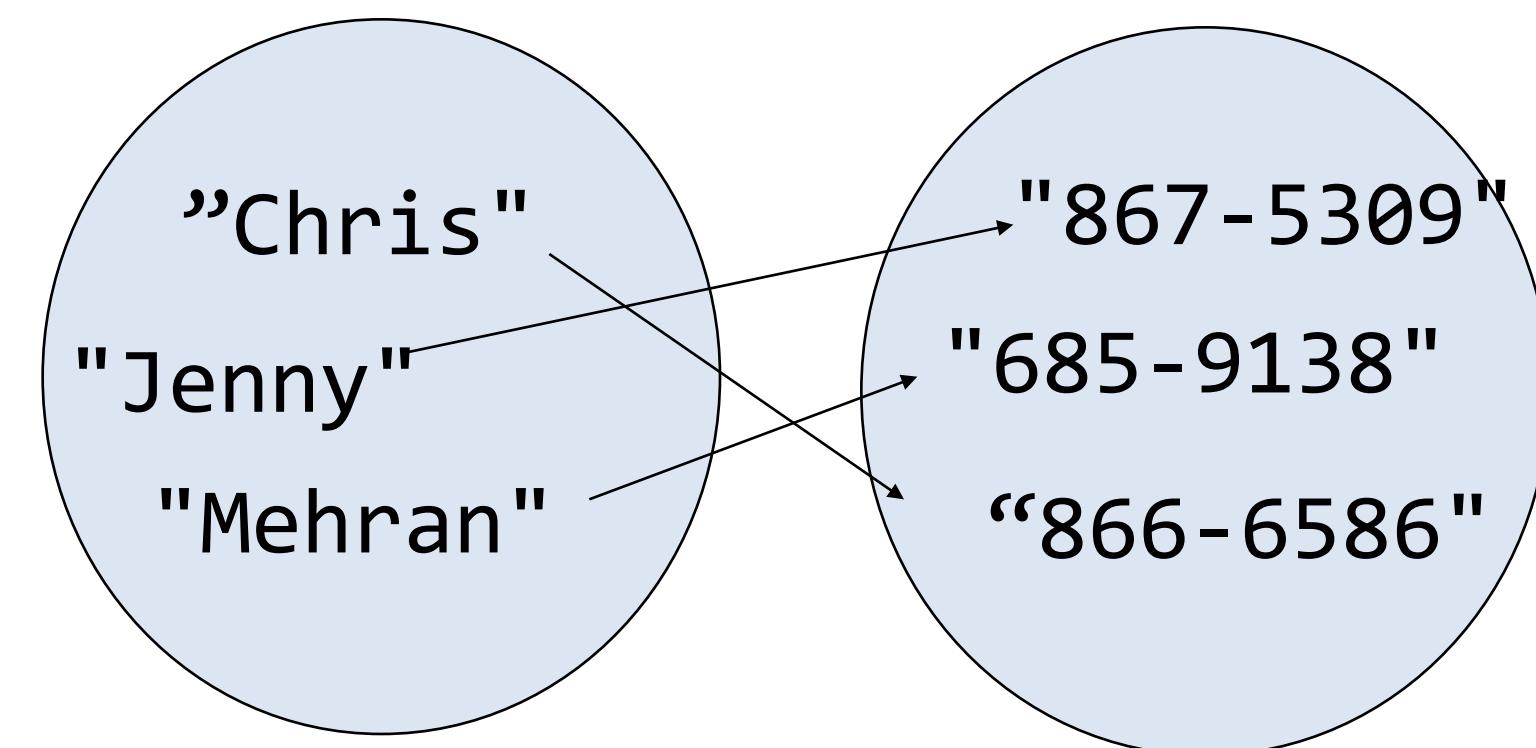
Sets

- Collection of elements with no duplicates



Maps

- Collection of Key/value pairs
- Use a key to find its associated value;



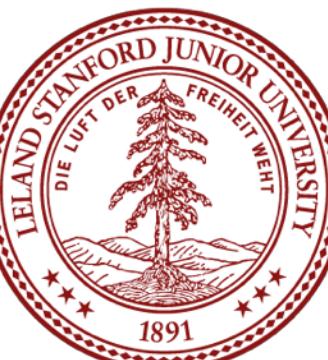
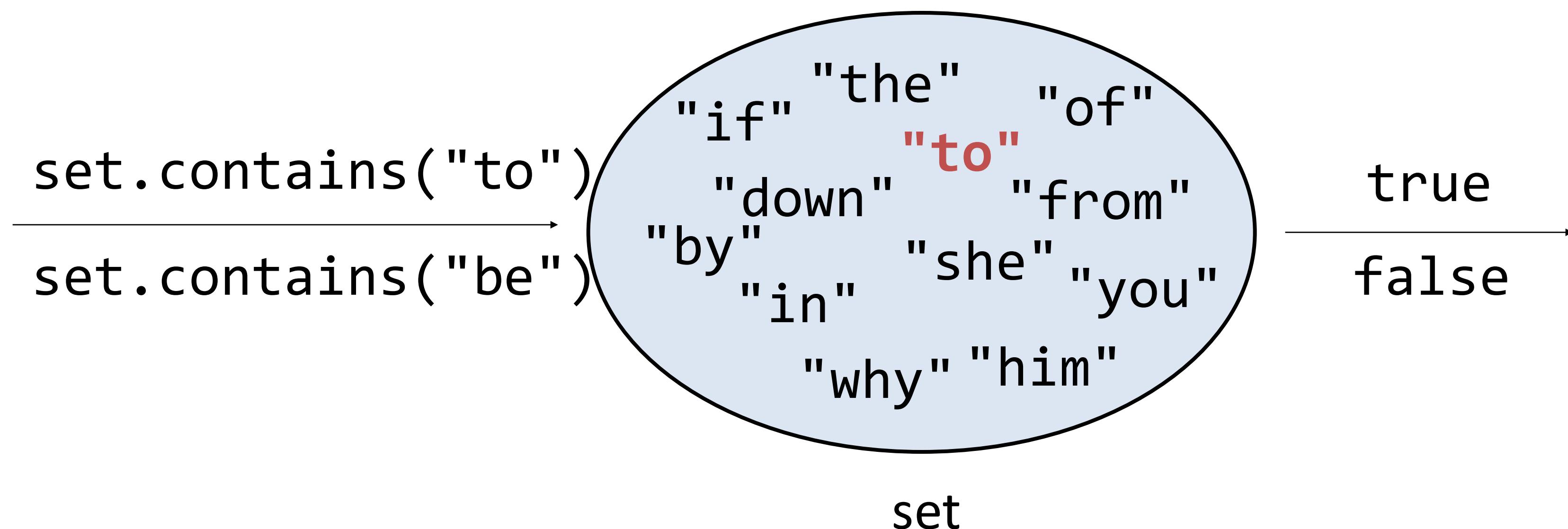
Both are blindingly fast to lookup an element



Sets

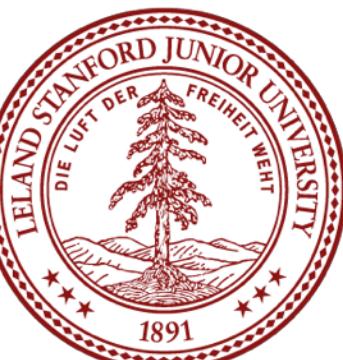
set: A collection of values with no duplicates.

Add, contains, remove operation are all fast
We don't think of sets as having indices



Simple Example

```
Set<string> friends;
friends.add("chris");
friends.add("anton");
cout << friends.contains("voldemort") << endl;
for(string person : friends) {
    cout << person << endl;
}
```



The Set Essentials

`set.size()`

Returns the number of elements in the set.

`set.add(value)`

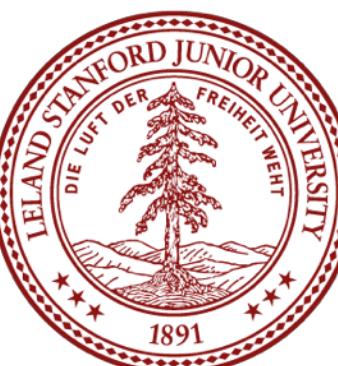
Adds a new value to the set (ignores it if the value already was in the set).

`Set.contains(value)`

Return true if the value is in the set.

Sets also have remove, isEmpty and other helpful methods.

See the online docs for more.



Looping Over a Set

```
for (type currElem : set) {  
    // process elements one at a time  
}
```

can't use a normal for loop and `get` each element [i]

```
for (int i = 0; i <  
     // does not comp:  
     cout << set[i];  
}
```



Types of Sets

Set

Iterate over
elements in
sorted order

Really Fast

Implemented using a
binary search tree

HashSet

Element order is
jumbled

Really,
Ridiculously
Fast

Implemented using a
hash table



Count Unique Words: Revisited



Set Operands

s1 == s2	true if the sets contain exactly the same elements
s1 != s2	true if the sets don't contain the same elements
s1 + s2	returns the union of s1 and s2 (elements from either)
s1 * s2	returns intersection of s1 and s2 (elements in both)
s1 - s2	returns difference of s1, s2 (elements in s1 but not s2)



Set Operands

```
Set<string> allStudents = loadStanford();
```

```
Set<string> left;
for(string student : allStudents){
    if(audienceLeft(student)) {
        left.add(student);
    }
}
```

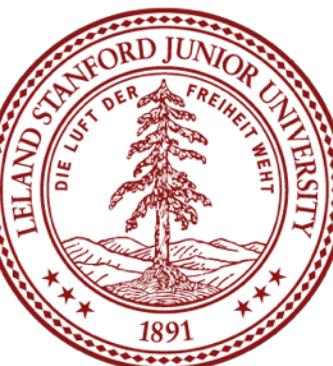
```
Set<string> hats;
for(string student : allStudents){
    if(isWearingHat(student)) {
        hats.add(student);
    }
}
```



Set Operands

allStudents, hats, left

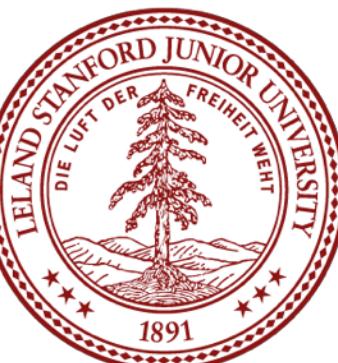
stand(allStudents);



Set Operands

allStudents, hats, left

stand(left);

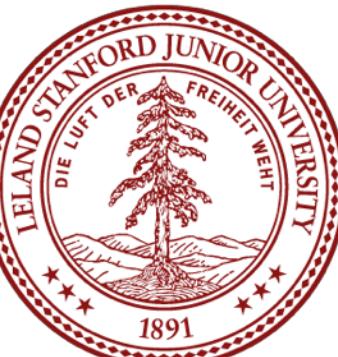


Set Operands

allStudents, hats, left

```
Set<string> combine;  
combine = left + hats;  
stand(combine);
```

Elements in
either set

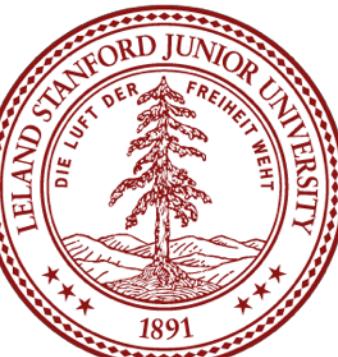


Set Operands

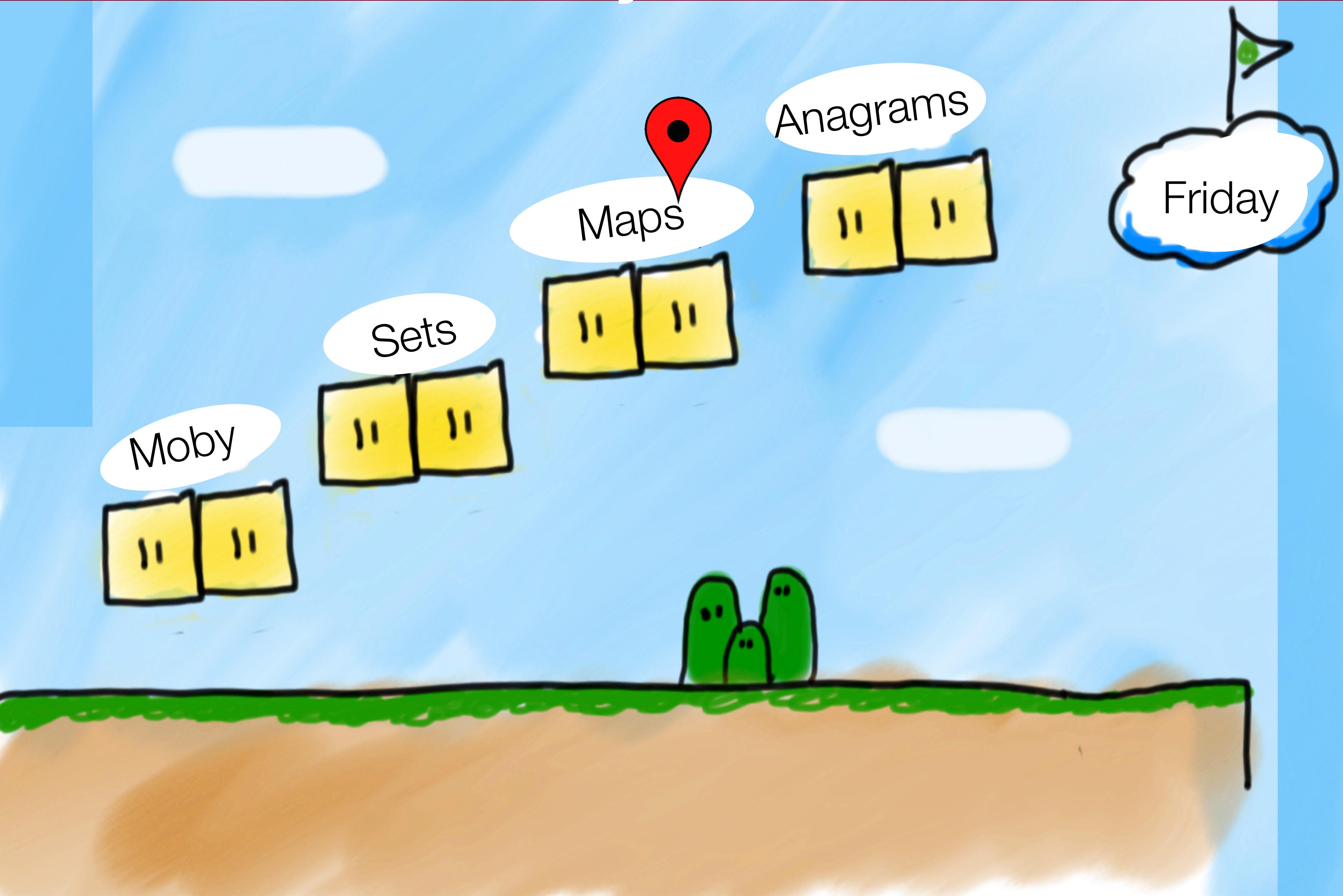
allStudents, hats, left

```
Set<string> combine;  
combine = left * hats;  
stand(combine);
```

Elements in
both sets



Today's Plan

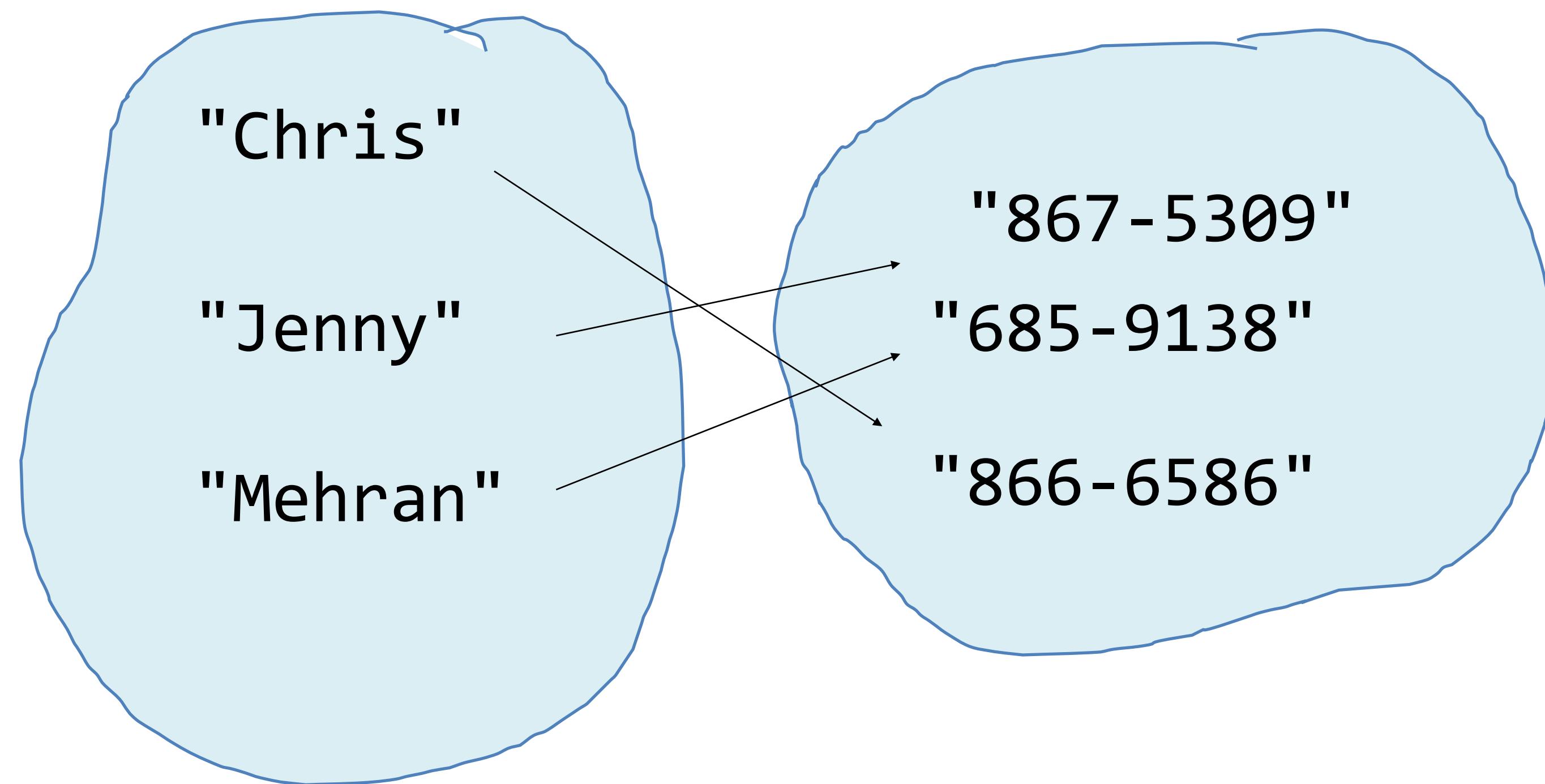


Maps

map: A collection of pairs (k, v) , sometimes called key/value pairs, where v can be found quickly if you know k .

a.k.a. dictionary, associative array, hash

a generalization of an array, where the "indexes" need not be **ints**.



Using Maps

A map allows you to get from one half of a pair to the other.

Store an association from "Suzy" to "206-685-2181".

```
//      key          value  
// m["Suzy"] = "206-685-2181";  
m.put("Suzy", "206-685-2181");
```



What was Suzy's phone number?

```
// m["Suzy"]  
m.get("Suzy")  
"206-685-2181"
```



Maps are Everywhere

Key = Title, Value = Article

Key:

“Mantis Shrimp”

Value:

Mantis shrimp

From Wikipedia, the free encyclopedia

Mantis shrimp or stomatopods are marine crustaceans, the members of the order Stomatopoda. Most species can grow to around 10 centimetres (3.9 in) in length, though a few species reach up to 38 cm (15 in).^[2] The largest ever caught has a length of 46 cm (18 in) in the ocean near Fort Pierce, Florida of USA.^[3] The carapace of mantis shrimp covers only the rear part of the head and the first four segments of the thorax. There are more than 400 species of Mantis shrimp. Varieties range from shades of brown to vivid colours, and are among the most important predators in many shallow, tropical and sub-tropical marine habitats. Despite being common, they are poorly understood as many species spend most of their life tucked away in burrows and holes.^[4]



Odontodactylus scyllarus

Scientific classification

Kingdom:	Animalia
Phylum:	Arthropoda
Subphylum:	Crustacea
Class:	Malacostraca
Subclass:	Hoplocarida
Order:	Stomatopoda
	Latreille, 1817

Superfamilies and families [1]

Bathysquilloidea

Key:

“Antelope Canyon”

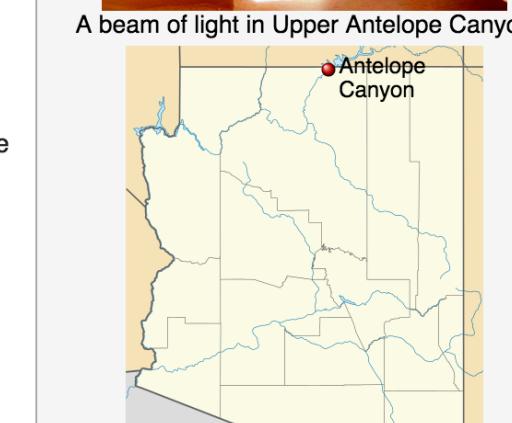
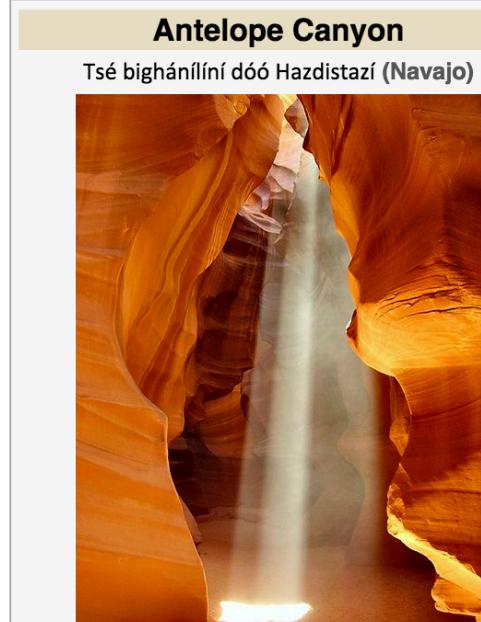
Value:

Antelope Canyon is a slot canyon in the American Southwest. It is located on Navajo land east of Page, Arizona. Antelope Canyon includes two separate, photogenic slot canyon sections, referred to individually as Upper Antelope Canyon or The Crack; and Antelope Canyon or The Corkscrew.^[2]

The Navajo name for Upper Antelope Canyon is Tsé bighánílíní, which means "the place where water runs through rocks." Lower Antelope Canyon is Hazdistazí (advertised as "Hasdestwazi" by the Navajo Parks and Recreation Department), or "spiral rock arches." Both are located within the LeChee Chapter of the Navajo Nation.^[4]

Contents [hide]

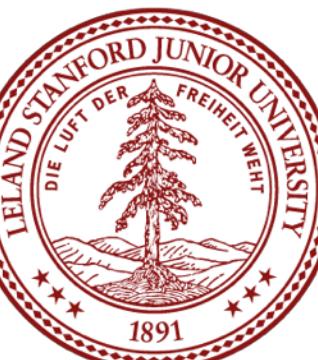
- 1 Geology
- 2 Tourism and photography
- 2.1 Upper Antelope Canyon



Creating Maps

Requires 2 type parameters: one for keys, one for values.

```
// maps from string keys to integer values  
Map<string, int> votes;  
  
// maps from double keys to Vector<int> values  
Map<string, Vector<string>> friendMap;
```



Map Methods

<code>m.clear();</code>	removes all key/value pairs from the map
<code>m.containsKey(key)</code>	returns true if the map contains a mapping for the given key
<code>m[key]</code> or <code>m.get(key)</code>	returns the value mapped to the given key; if key not found, adds it with a default value (e.g. 0, "")
<code>m.isEmpty()</code>	returns true if the map contains no k/v pairs (size 0)
<code>m.keys()</code>	returns a Vector copy of all keys in the map
<code>m[key] = value;</code> or <code>m.put(key, value);</code>	adds a mapping from the given key to the given value; if the key already exists, replaces its value with the given one
<code>m.remove(key);</code>	removes any existing mapping for the given key
<code>m.size()</code>	returns the number of key/value pairs in the map
<code>m.toString()</code>	returns a string such as " <code>{a:90, d:60, c:70}</code> "
<code>m.values()</code>	returns a Vector copy of all values in the map



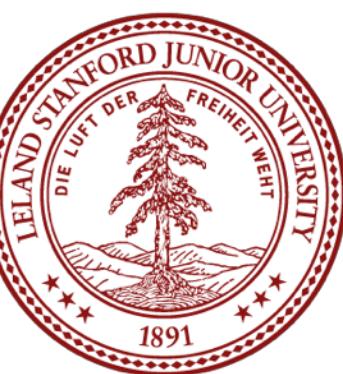
Map Example

```
Map<string, string> wiki;

// adds name / text pair to dataset
wiki.put("Stanford Band", articleHTML);

// returns corresponding articleHTML
cout << wiki.get("Mantis Shrimp");

// removes the article
wiki.remove("Britain in the E.U.");
```



Types of Maps

Map

Iterate over
elements in
sorted order

Really Fast

Implemented using a
binary search tree

HashMap

Element order is
jumbled

Really,
Ridiculously
Fast

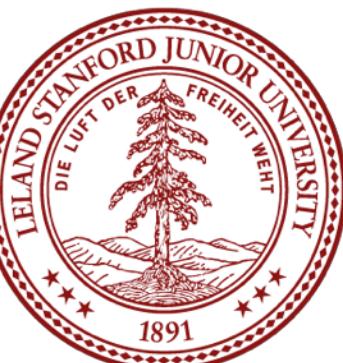
Implemented using a
hash table



Tallies

count digits: 22092310907

index	0	1	2	3	4	5	6	7	8	9
value	3	1	3	0	0	0	0	1	0	2

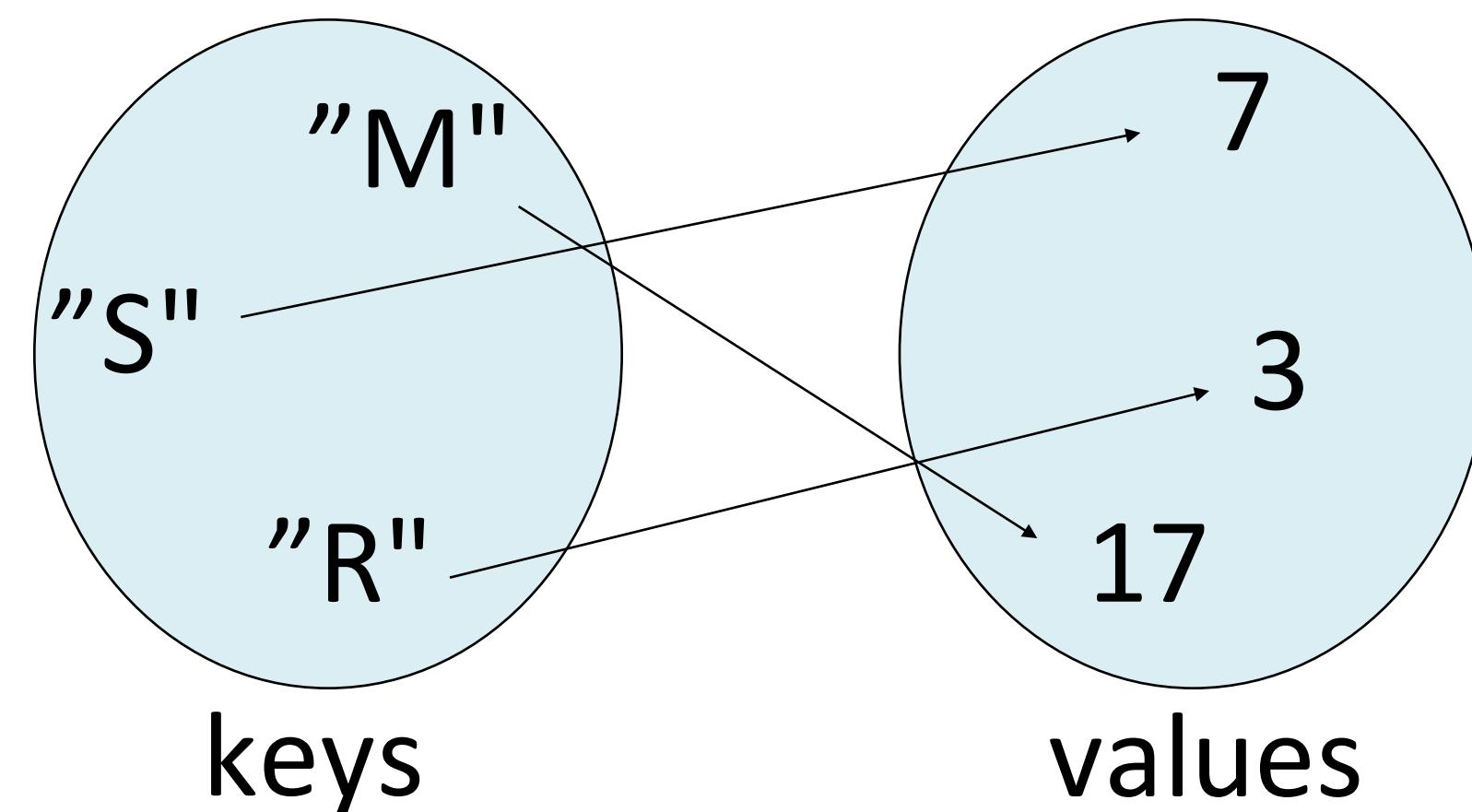


Tallies

count votes:

// (M)ilk, (S)tokes, (R)ogers
“MMMRMSSMSMRRMMRMRRMM”

key	"M"	"S"	"R"
value	17	7	3



*In 1976 Harvey Milk became the first openly gay elected official in the US



Problem #2: Tally Words



Looping Over a Map

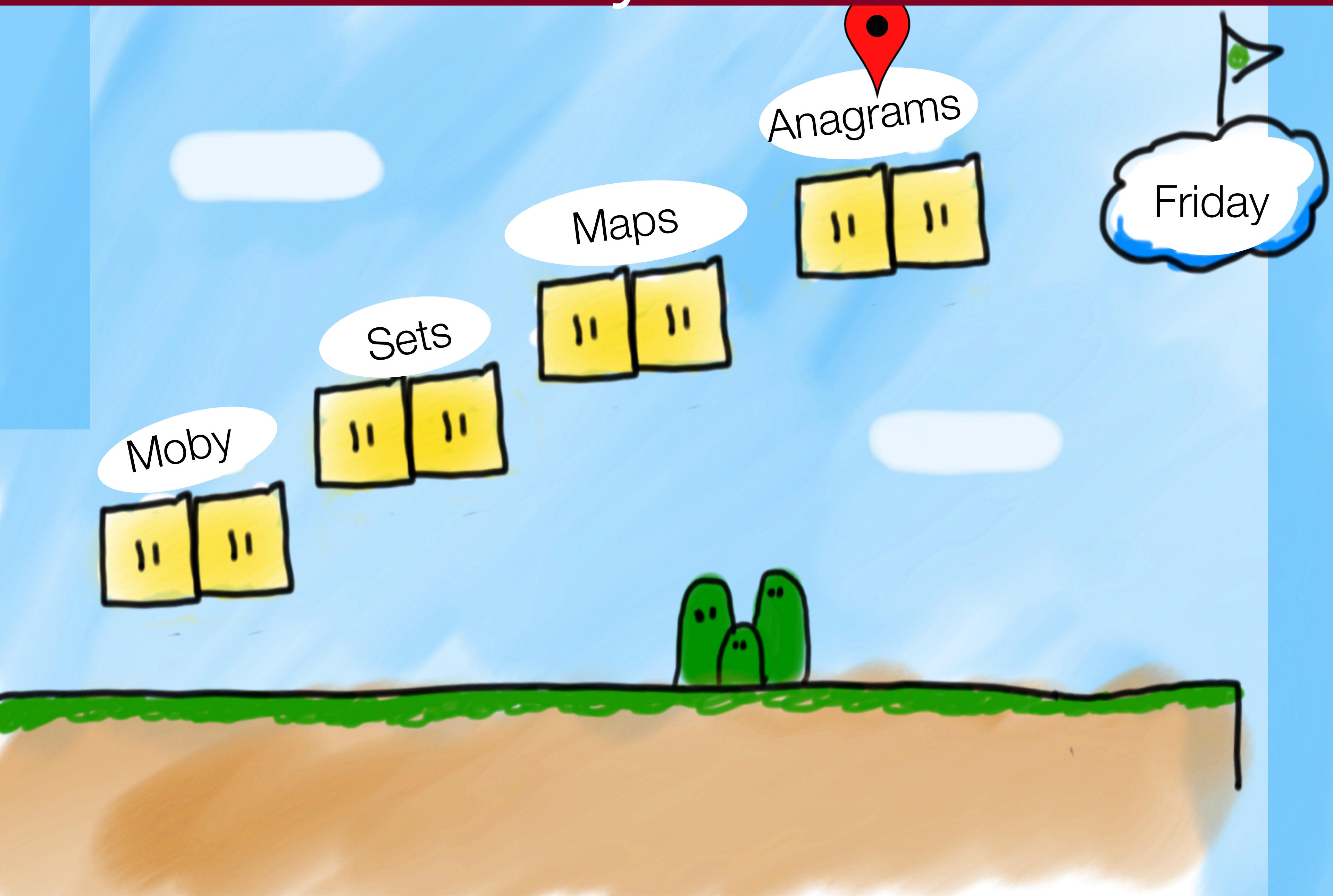
```
Map<string, double> gpa = load();

for (string name : gpa) {
    cout << name << "'s GPA is ";
    cout << gpa[name] << endl;
}
```

*The order is unpredictable in a HashMap



Today's Plan





Sets, good

Maps, good

Sets + Maps, good

Anagram Problem

Write a program to find all anagrams of a word the user types.

Type a word [Enter to quit]: scared

Anagrams of scared:

cadres

cedars

sacred

scared

What is the appropriate collection to use to solve this problem?

Hint: Use a compound collection...

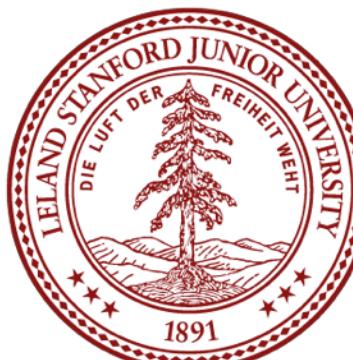


Insight

Every word has a sorted form where its letters are arranged into alphabetical order

word	sorted form
- fare	-> aefr
- fear	-> aefr
- swell	-> ellsw
- wells	-> ellsw

Notice that anagrams have the same sorted form as each other



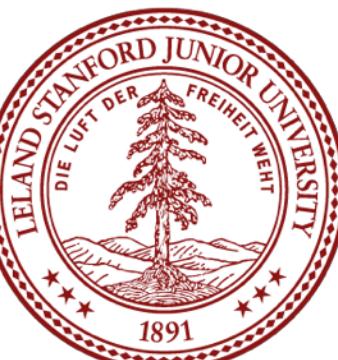
Lexicon: A set of vocabulary

Lexicon: Special class for sets of strings

- Often used to store vocabularies
- Great syntax for loading from file.
- Same methods as set.
- Also has a **containsPrefix** method.



Historically the Lexicon was created for file size reasons (3MB was too large)



Lexicon Example

```
// great syntax for loading from file
Lexicon english("english-dictionary.txt");

// true: hello is a word
cout << english.contains("hello") << endl;

// true: hel is not a word but its a prefix for hello
cout << english.containsPrefix("hel") << endl;

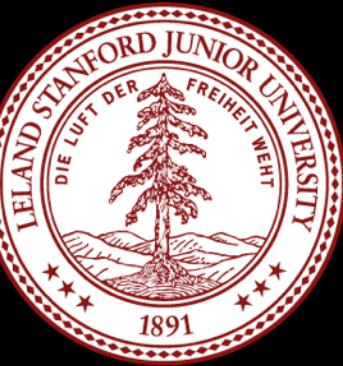
// same syntax for looping over all elements as set
for(string word : english) {
    cout << word << endl;
}
```



Problem #3 Anagrams



Aka how to beat your friends at word games

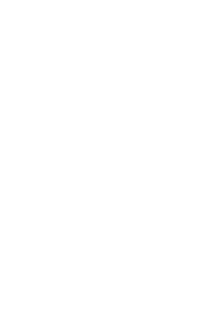


Me Making this Lecture...

I'm really into this anagram solution. It feels like it can be used for so much more...



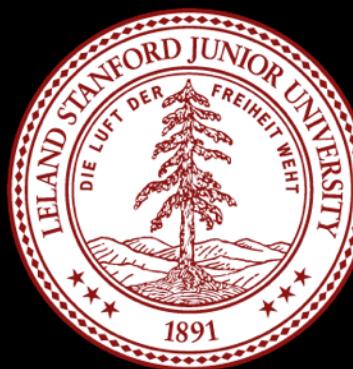
Bananagrams artificial intelligence



Bananagrams



<http://mysummeratamonastery.blogspot.com/2013/07/cel-rotis-stolen-money-and-bananagrams.html>





Bananagrams vs computer



Bananagrams online



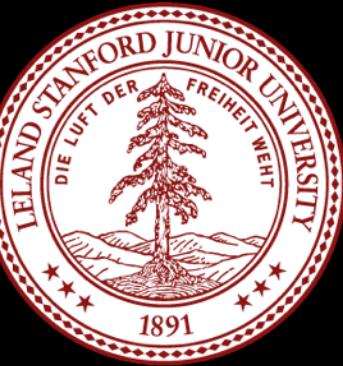
number of legal moves in bananagrams



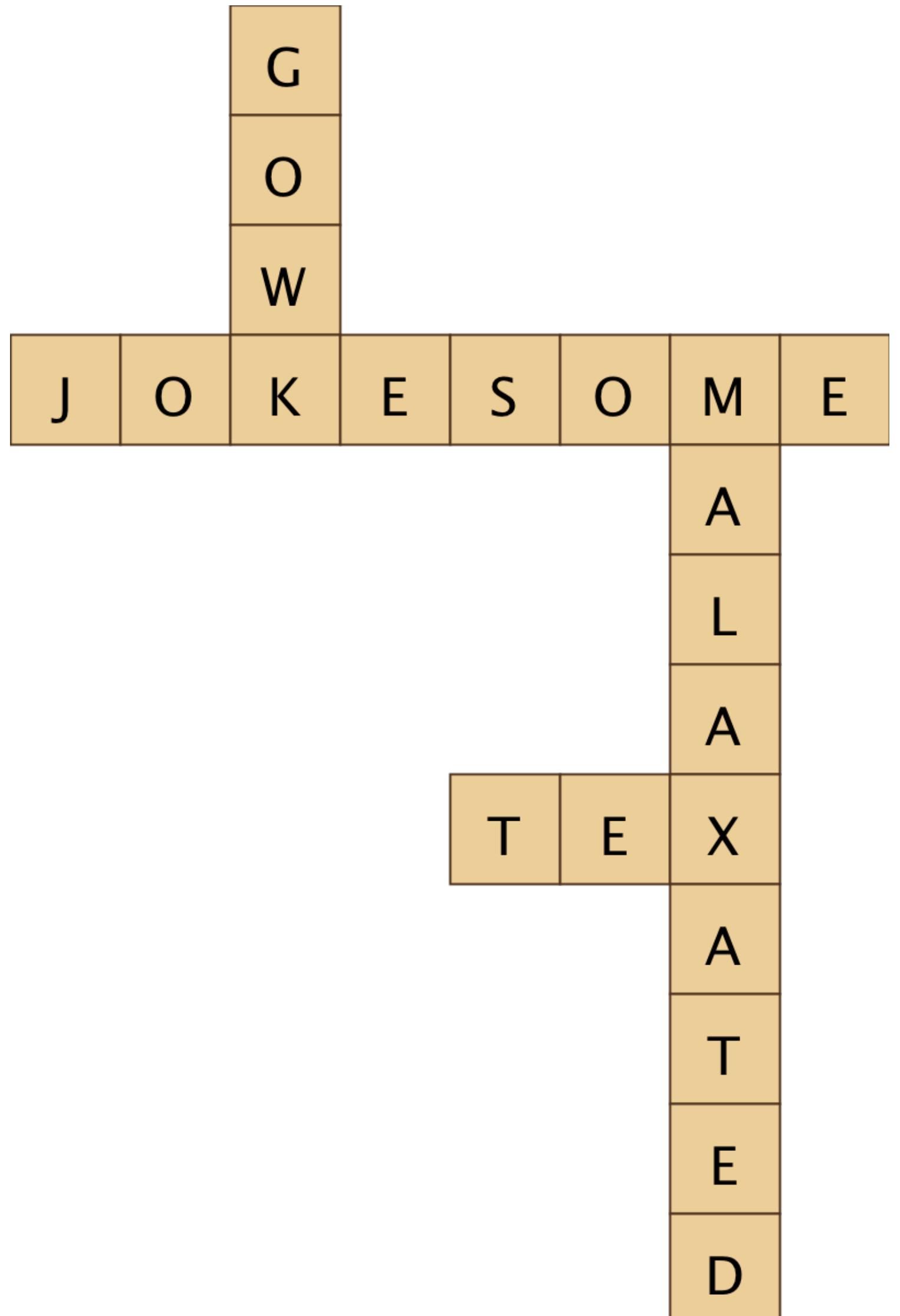
different species of bananas



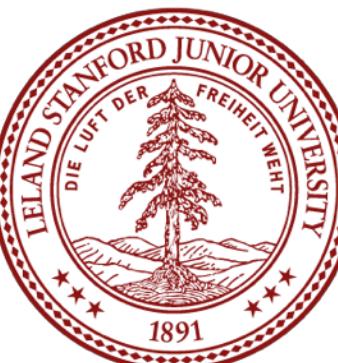
This may be the first...



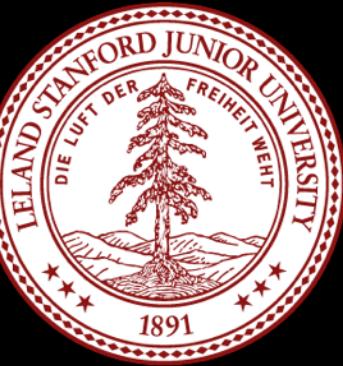
Bananagrams



Tiles: DTGEMAOTWKJEAAESELXO

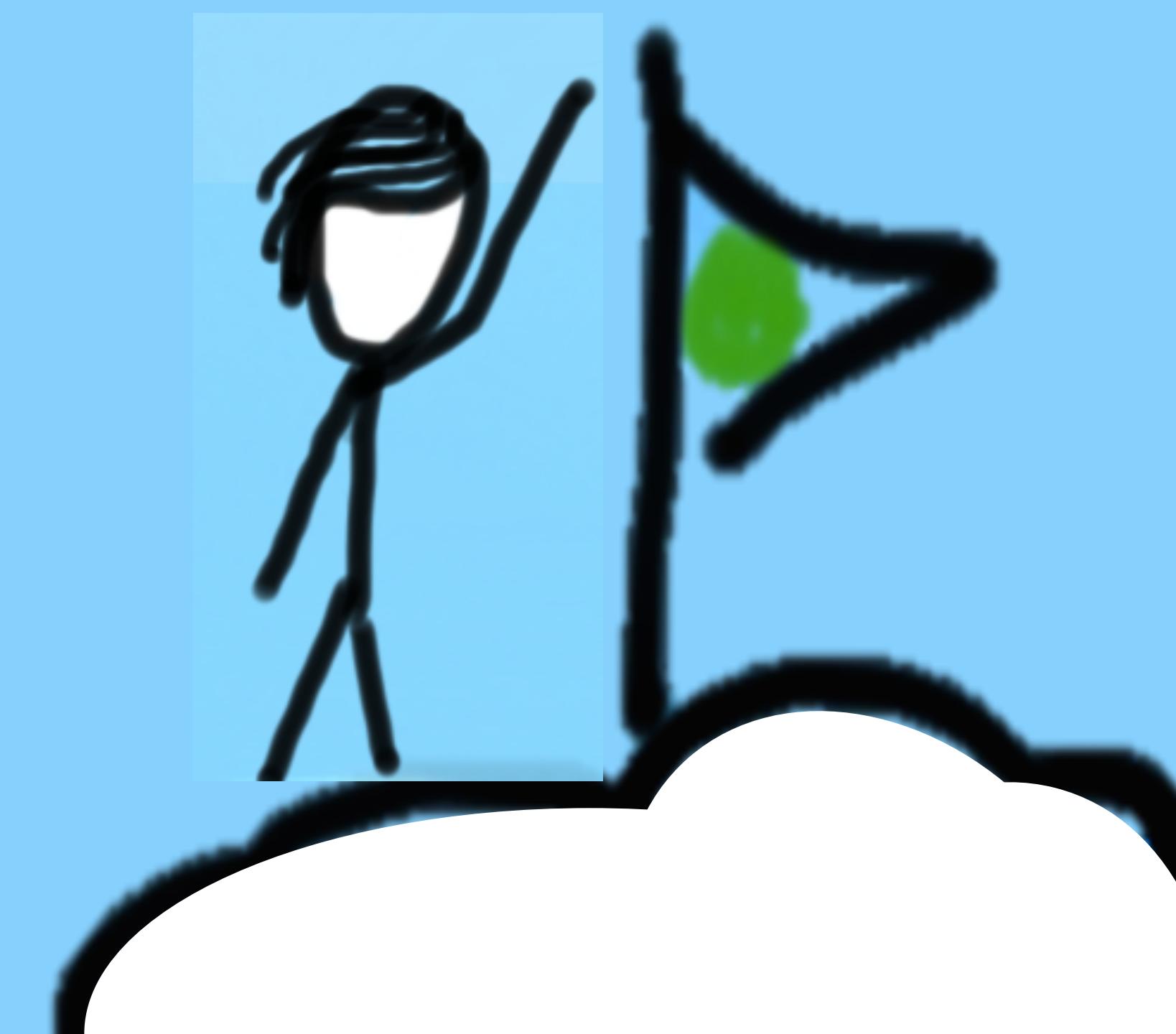


Bananagrams code online

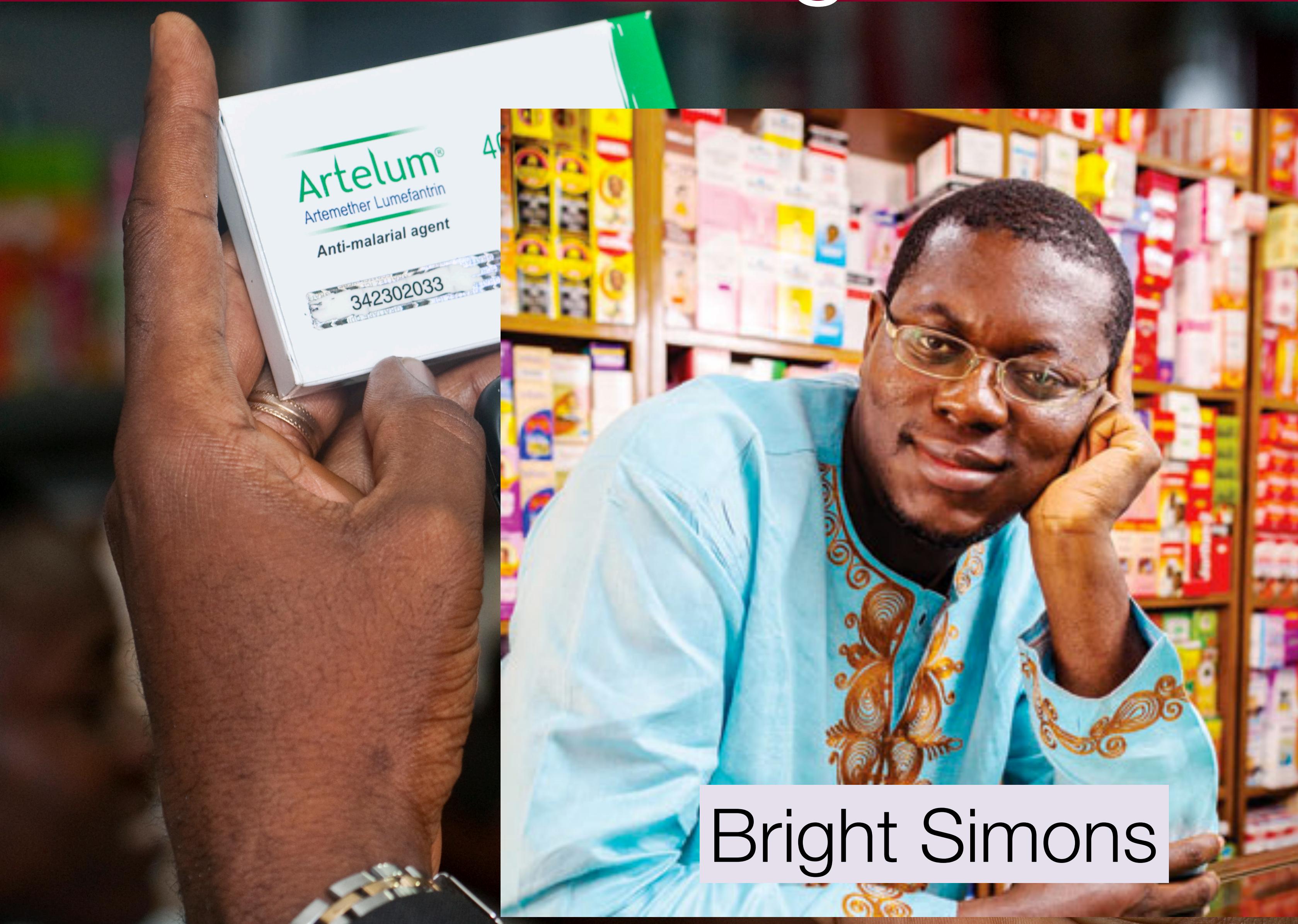


Today's Goal

1. Learn how to use Sets (and Lexicons)
2. Learn how to use Maps
3. Be prepared for Assn 2



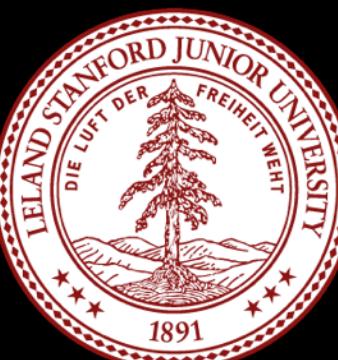
M Pedigree



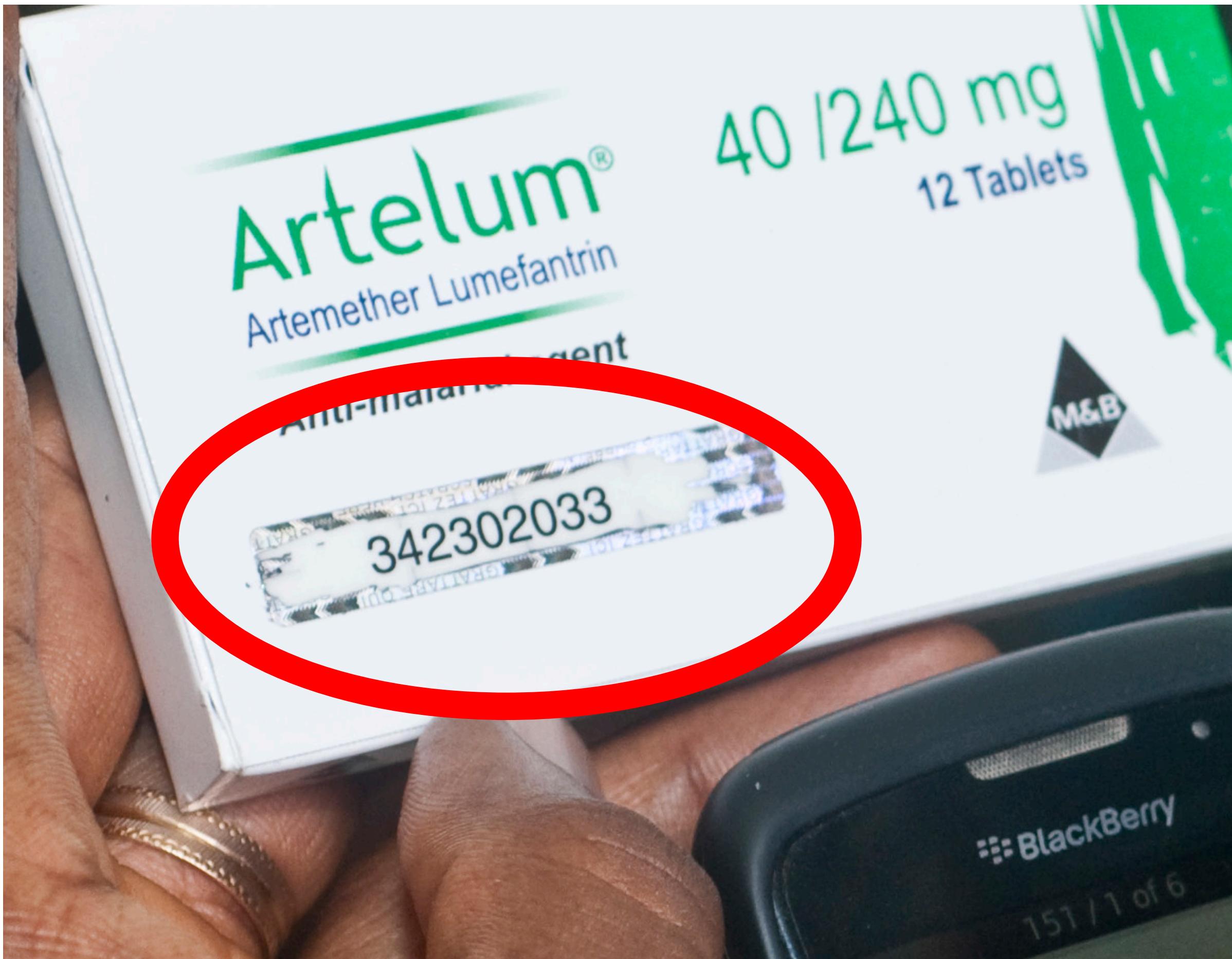
Bright Simons

What Data Structure?

- A. Vector<int>
- B. Set<int>
- C. Map<int, string>
- D. Stack<int>



M Pedigree



342302033

“legitimate”

927216398

“used”

⋮

927216398

“legitimate”



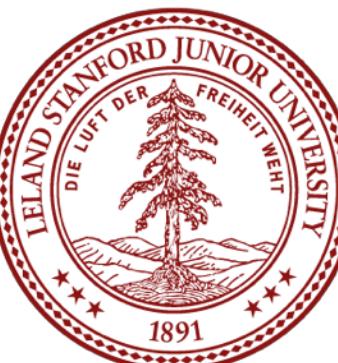
M Pedigree



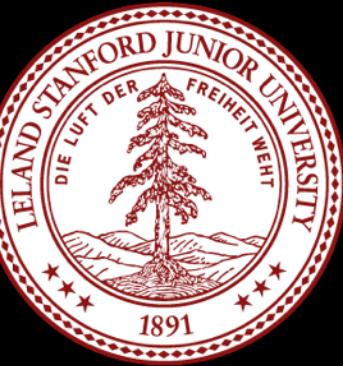
Random number

34230 2033

Unique Id



Happy Friday



Extra Reading

Hashing: https://en.wikipedia.org/wiki/Hash_table

Relational Databases: https://en.wikipedia.org/wiki/Relational_database
(especially indices)

